Transcribed from the Disk Magazine

COMPUTE!'s GAZETTE

March 1994
Disk Issue No. 03

FOR **COMMODORE** PERSONAL COMPUTER USERS

Transcription prepared for the CBM 8-Bit Magazine Index by Dr. Myles L. Skinner. Please visit https://www.tierceron.com/commodore/magazine_index.php for more information about the index.

GAZETTE DISK MASTHEAD

Tom Netsel, editor
Address editorial matters to the Greensboro office or via E-mail: Gaz on QuantumLink, TomNetsel@aol.com on the Internet

Sylvia Graham, copy editor

Matthew Byrd, intern

Editorial offices:
COMPUTE's Gazette
324 W. Wendover Ave., Ste. 200
Greensboro, NC 27408
(910) 275-9809

Subscription information:
Gazette Disk
P.O. Box 3244
Harlan, IA 51593
(800) 727-6937
Call this number for any questions about your subscription, change of address, missing issues.

Advertising information:
Lucille Dennis
(707) 451-8209

Administration:
Kathy Keeton, president and COO
Bill Tynan, executive vice president operations
COMPUTE Publications Intl. Ltd
1965 Broadway
New York, NY 10023
(212) 496-6100

All material on this disk, with the exception of public domain programs featured in "PD Picks," is copyrighted 1994 by COMPUTE Publications Intl. Ltd.

Cover image created by Chris Ryan using *The Print Shop* from Brøderbund Software and originally printed on a Panasonic KX-P1180 9-pin dot-matrix printer.

Transcription prepared for the CBM 8-Bit Magazine Index and edited by Dr. Myles L. Skinner (MS). Please visit https://www.tierceron.com/commodore/magazine_index.php for more information about the index.

(The Gazette Disk *masthead doesn't appear until Disk Issue No. 05, but including it here seems like the best way to give proper credit to those who deserve it. –MS)*

Table of Contents

64/128 View
Tom Netsel 3

Feedback 4

Advertising 66

Columns

GEOS: The Lighter Side
Steve Vander Ark 7

Machine Language: BASIC Fun
Jim Butterfield 10

Beginner BASIC: Lessons
Larry Cotton 12

PD Picks: Pegasis and Ghoul Dogs
Steve Vander Ark 15

D'Iversions: The Real MPCs: Kids Inventing Multimedia on Their Own
Fred D'Ignazio 17

Programmer's Page: SpeedScript Programming
Dave Pankhurst 20

Features

Review: Full Count Baseball
Don Radler 24

Commodore Buyer's Guide
Lisa Hayes, comp. 26

SpeedCalc
Kevin Martin 40

Programs

SpeedScript Integrator
Kevin Martin 51

Batch Programming
Steven Reid 51

Setz!
Larry Cotton 62

Trail Blazer
Gus Vakalis 63

StereoGram 128
Edward Sullivan 64

Pegasis
Cleveland M. Blakemore 65

Ghoul Dogs
Melvin Finstemocker 65

64/128 VIEW

Tom Netsel

Sorry about making you flip disks right off the bat while reading the columns, but I ran out of space on the front side. This is the last file that usually goes on the disk and the only vacancy for it was on the back side.

Some months ago I made a request for templates that would work with *SpeedCalc,* our spreadsheet program for the 64. A template is a file that loads into a spreadsheet and performs a certain chore, such as calculating a budget, tracking stock portfolios, figuring interest payments, and so on. I received some excellent submissions, but I also had quite a few letters asking about *SpeedCalc* itself and when it was published in *Gazette.*

Since *SpeedCalc* never appeared in *Gazette* (it was published in *COMPUTE* back when that magazine contained program listings for various computers), I decided to offer the program to *Gazette* readers. You'll find *SpeedCalc* ready to run on this disk. The documentation is quite large, but it provides a lot of useful information that applies to spreadsheets in general. So if you've never used a spreadsheet before, here's your chance.

SpeedCalc is a powerful program despite its small size. It offers you a number of advanced mathematical functions, formats, screen and text colors, and the like.

SpeedCalc can handle a great many of your spreadsheet chores. One beauty of this type of program is that its power is limited only by your mathematical prowess—just as a word processor is limited only to your skill with words.

If you find that you need a spreadsheet with more advanced functions and features, you may want to consider *GemCalc.* This spreadsheet is on the *Gazette Productivity Manager Disk* ($16.95). *GemCalc* was written by talented Australian programmer Tim Strickland, and is based on *SpeedCalc,* using many of the same commands and formulas. *GemCalc* offers onscreen help, sample files, more flexible sum and averaging of data blocks, and more sophisticated column and row manipulation. There are versions for both the 64 and 80-column 128.

In looking back at the original *COMPUTE* article in which *SpeedCalc* appeared, I found a utility program called Integrator. This program lets you take your spreadsheet files and convert them into a format that you can load into *SpeedScript, COMPUTE's* word processor. This ability lets you to print spreadsheet files using *SpeedScript's* different fonts, centering, and formatting capabilities that are not available with *SpeedCalc* when you want to print a file.

Speaking of *SpeedScript,* did you know that you can use that word processor to read and print any of the text files that you find on *Gazette Disks?* As a matter of fact, I'm writing this article with *SpeedScript* and after editing, I'll save it to this month's master disk. As you know, we include a text reader on each disk that lets you read and print text files, but you can use *SpeedScript* to load and print them if you prefer.

Getting back to *SpeedCalc* for a moment, I feel a little guilty about putting this program on disk when I have so many other good programs waiting to be published. Your submissions are starting to back up again and I ask your indulgence while I go through them all. I hope you progammers have seen the ad on the disk that lets you know that *Gazette* is still interested in buying your best programs.

I'm also on the lookout for feature articles and tutorials. If you can write and explain some aspect of the 64 or 128 in a way that might entertain and inform our readers, drop me a note outlining your proposed article.

Some of you new subscribers may be seeing *Gazette Disk* for the first time, but we've produced a companion disk for each issue of *Gazette* for almost a decade. You may not know it but none of our programs are copy protected. You can load and save them to your own work disks, make back-up copies, or whatever. Bear in mind that the programs are copyrighted, so we ask that you not sell them or give copies away.

I hope you'll let me know what you like and don't like about the disk. Last month we ran a huge program for the 128, but this month I notice that we have only one program on the 128 menu. Do you want more 128 programs? Do you want more games or utilities? How about graphics or musical selections? Do you want more golden oldies from *Gazette's* and *COMPUTE's* past or do you want only new, previously unpublished programs? Your feedback will be appreciated.

If you are reading the *Gazette* Columns now, remember to flip the disk back to the front side before you press the M key for the other column selections. This is the only column on this side. If you make a selection and get a Disk Error message, you're probably on the wrong side of the disk. Just flip it over and press Return.

Feedback

Bug-Swatter

If you try to run *Unbreakable Cipher* from the menu of the November 1993 *Gazette Disk*, the program crashes. The program is much too large to use with the menu program as it stands. To fix it, load MN, the table of contents file, and enter the following changes.

```
470 GOTO 550
```

List line 550 and change its line number to 551. Then enter

```
550 IF C=3 THEN POKE 55,0: POKE 56,160
```

Scratch MN and save this new version with the same filename. (Note: you'll have to make a notch in the disk in order to write to the original disk. It's safer to make a copy of the disk and make any corrections on the copy.)

This change does not reflect a generic problem with MN. In line 30, the top of BASIC is lowered to 14336 to protect the font, ITALICS.SET, which has a starting address equal to

this value. Since *Unbreakable Cipher* is too large to fit into the remaining memory, the top of BASIC has to be raised to 40960 before the boot file, CIPHER BOOT, is loaded and run. Otherwise, you get an OUT OF MEMORY error in line 100.
JAMES T. JONES
KLONDIKE, TX

James Jones is the author of Unbreakable Cipher.

Reader to Reader Help

I was reading an old magazine and found an article that mentioned a program called *Mind Prober.* It was published by Human Edge Software. I have contacted a number of software distributors but no one has this program. Can any of your readers help me locate this program?
GARY L. REYNOLDS
5213 FRONT ROVER RD.
PITTSBURGH, PA 15225

64 Emulator

If one purchases an Amiga 500 or 600, is there any way that BASIC programs, programs such as *MetaBASIC,* and commercial software for the 64 can be used on them?
LEE NORRIS
KEARNEY, NE

Questronix (P.O. Box 340265, Hartford, Connecticut 06134) puts out a product called The A64 Package *that is a full-featured 64 emulator for all Amigas. It sells for $69.95. Look for a review of the product in coming issues of* Gazette Disk.

Lower the Checksum

Is there any way to lower the checksum for the *Automatic Proofreader?* The top half is cut off by the television frame and I can't be certain what the correct checksum really is when I type in programs.
LEWIS DICKERSON
HILTON, NY

Readers using The Automatic Proofreader *on a 64 can enter POKE 53265,31 in immediate mode to lower the checksum's position on a TV screen. Be aware that this will also lower the position of the last line on the screen.*

We still include The Automatic Proofreader *and* MLX *on this disk for those readers who may want to type in some programs from earlier magazines.*

New *Gazette* Format

I am writing to wish you good luck in your endeavors with the new *Gazette* format. I hope that this restructuring will allow *Gazette* to continue to bring Commodore information into my home for a long time to come.

I have owned a 64 for ten years and presently on three of them; I sold a fourth to my sister. I have a large library of software that I use regularly. My wife and three children use it more and more all the time. My children have no desire for a Nintendo or Sega as their computer lets them play games and write reports, create invitations, and so on.

One of the first things I bought for my 64 was the December 1983 issue of *Gazette.* I have been a subscriber to the magazine ever since. I typed in *SpeedScript* when I didn't own a printer, and then I ordered disks whenever typing became too much. I have seen *Ahoy!, RUN, geoWorld,* and *geoVision* all come and go. I hope you can keep up the good work.

I called and changed my subscription and renewed it at the same time. I look forward to the handy new format and less typing. Thanks once again for all the news you have brought me, useful programs, and reviews. I look forward to many more issues of *Gazette* in the future.
BRUCE THOMAS
EDMONTON, AB
CANADA

At first, I was skeptical about the new disk. I am used to the convenience of a printed magazine. Magazines are portable and one can tell at a glance the size of an article, but I'm pleased with the new disk. It seems to be tailored to the 64 and that's a plus. I'm not interested in IBM.
STAN POWELL
SUMMERVILLE, SC

I think switching *Gazette* from a magazine to a disk product stinks. I like the feel of flipping through pages of a magazine. I can't browse through a disk the same way. Sorry, but I am dropping my subscription.
MUNDO DURAN
NEW YORK, NY

Disable the RUN/STOP Key

I am 12 years old and I like to program my 128. Can you tell me how to disable the Run/Stop key?
JAMES DURRANT
PAYSON, UT

On the 64 you can disable the Stop key with POKE 808, 239. You can disable the Run key with POKE 808, 127. POKE 808,237 to restore any of these pokes.

On the 128, you can disable the Stop key with POKE 808, 103. To restore it, POKE 808, 110.

Back Issues of *Gazette*

Is it possible to receive a listing of available back issues of *Gazette* and their contents?
JANET SZUCS
BRADFORD, ON
CANADA

At this time, we don't have the listing that you ask for, but most older issues of the disk and magazine are still available. If you have the Gazette Index, *you can find information about many of our programs and articles*

and the dates that they were published. Any back issue of the magazine costs $6.00 and can be ordered from our editorial office in Greensboro.

Back issues of the Gazette Disk *vary somewhat in price. Any disk from the present back to October 1990 costs $9.95 plus $2.00 shipping, but disks from June 1990 and earlier are real bargains. Those disks cost $3.50—and that includes shipping. Please remember, however, that prior to January 1994, instructions for using the programs were not on disk; they were in the magazine only. So you'll need the magazine to get the most out of the programs.*

To order back issues, write to:
Gazette Single Disk Sales
COMPUTE
324 West Wendover Avenue, Suite 200
Greensboro, North Carolina 27408

Hurricane Update

In response to Lou Rosen's request for a hurricane tracking map (November 1993 "Feedback"), I have one that is public domain. It is available from Disks O'Plenty, 8362 Pines Boulevard, Suite 270, Pembroke Pines, Florida 33024. The company sells public domain and shareware programs which are not expensive.
TYMOTHY MITCHELL
ALEXANDRIA, LA

Thanks for the nice plug for one of our advertisers.

If you have a question or comment, or need help from our readers, write to:
Gazette Feedback
COMPUTE
324 W. Wendover Ave., Ste. 200
Greensboro, NC 27408

GEOS

Steve Vander Ark

The Lighter Side

GEOS is the premier productivity software for the 64 and 128. *geoWrite* and *geoPaint* are among the very best programs of their types, and *geoPublish* is second to none in the desktop publishing category. The fact that these programs work together, along with a host of utilities and add-on programs, makes GEOS by far the best way to be productive with your Commodore.

I realize that someone might start extolling the virtues of *Word Writer* or pointing out the shortcomings of *geoFile* as compared to *Superbase.* Even so, GEOS still wins. Once you take into account the entire GEOS environment, those other programs can't compete. The environment's the key.

Let me explain what I mean by GEOS environment. On one level, I'm referring to the fact that you can share information—graphics, word processor text, spreadsheet data, and so on—between GEOS applications. You can, for example, copy a graphic image from *geoPaint* and paste it into *geoWrite.* You can use *geoMerge* to merge data from *geoFile* into a series of letters printed from *geoWrite.* This is possible because each application follows the same basic rules for creating and handing each kind of data. This is an extremely powerful capability.

At another level, the GEOS environment means that everything works the same way. The basic functions of GEOS, such as dialog boxes and pull-down menus, are identical or very similar on the *deskTop* and in every application. This kind of consistency makes GEOS surprisingly easy to master, no matter which applications you use. If, for example, you want to open a new document, you always pull down the File menu in any application.

I realize that I'm not revealing any startling tidbits of information here. I'm just quoting all those cool multipage GEOS advertisements Berkeley used to run in *Gazette* back in the 1980s. In fact, the letter E in GEOS stands for Environment. (You do know that GEOS stands for Graphic Environment Operating System, don't you?)

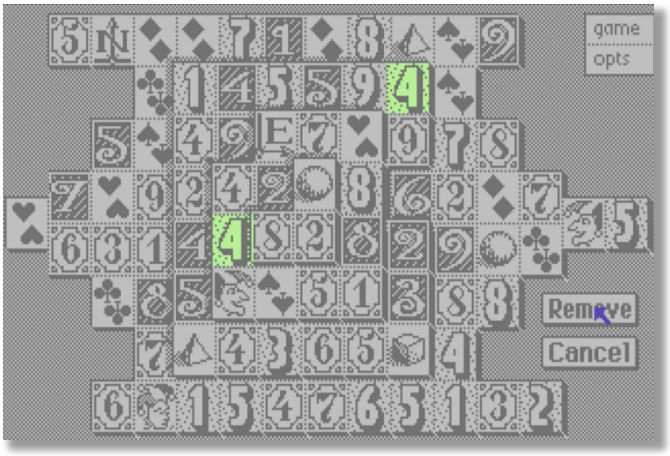
But there's another way of thinking of the GEOS environment, a much more imaginative way. An environment is a place, really. It's a space and all the things in it and how they interact. With this in mind you can think of GEOS as being just one large multifaceted workspace, sort of like a large office complex with different jobs being done in different areas. In this analogy the *deskTop* is like a central lobby which provides connections to each of the various work areas. *geoWrite* and *geoPaint* are a couple of those areas, each a specialized work space with the tools needed for one type of job. Each of these main areas also has a number of smaller sections as well, places where support work for that application can be done. Those are utility programs such as pattern editors for *geoPaint* or word counters for *geoWrite.*

Let's run with this idea for a minute. Like any modern office complex, our GEOS environment is interconnected with ways to communicate, to transfer information, and to take care of maintenance tasks. GEOS has the photo and text managers, the calculator, and numerous other desk accessories that reach neatly into every work area. They provide a connection from within an application out to the larger GEOS complex and into other applications. The whole thing is designed to make each kind of work in any area as efficient and, when necessary, as interconnected as possible.

Besides all those main work areas, any office complex needs a general work area for getting nonspecific jobs done. In our GEOS environment that means things like converting graphics into GEOS format from outside sources. There also has to be a maintenance area, a place to keep the mops and screwdrivers and extra light bulbs. In GEOS, that's where you'll find *SuperValidate* and *geoLogger.*

I know this sounds a bit off the wall, but consider it. When you think of it this way, you can't help but see why GEOS is so effective as a productivity tool. No other single program, even if it has a few more features, is going to be able to do what GEOS can do over the long haul. Add to that the fact that GEOS offers a quick and intuitive way to handle files and disks, and you get a combination that is unbeatable. This concept makes so much sense, in fact, that it's used on most every computer in the world, either as Windows on the IBM or on the Macintosh as, well, the Macintosh.

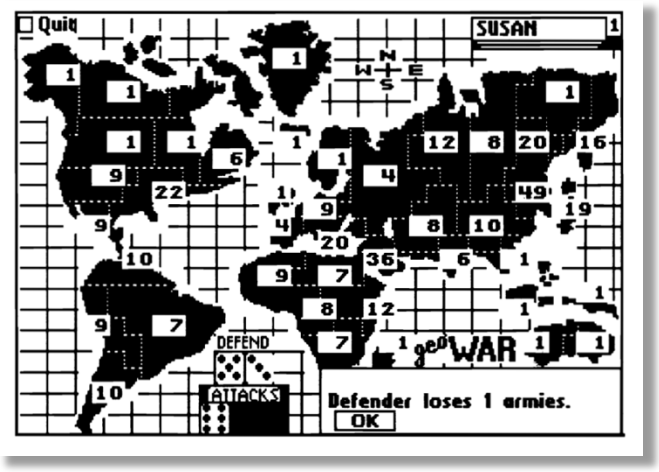
Lest I be accused of wasting an entire column trying to convert people who already believe, let me justify this exercise in philosophy by heading to the part of the office complex that I like best: the lounge. I figure any good office complex has to have a coffee pot, a deck of cards, and something like a rude-fax contest. Sure enough, the GEOS environment has its lighter side. So here are a few of my favorite ways to waste precious productivity time.



geoTile by Francis Kostella is available on *GEOS PowerPack Disk #2* from Creative Micro Designs.

This is my favorite GEOS game of all time. It's a version of the very popular mah-jongg game that originated ages ago in the Far East and has seen new and vibrant life as a computer game. I have several versions of mah-jongg for my Commodore and for my other computers, but I like it best in GEOS. I'm not sure why this version has so charmed me, unless it's that nifty animation that breaks in on the screen when you win. I don't dare leave *geoTile* on my regular work disks, or I'd never get to work!

geoWar by Lysle E. Shields III can be found on QuantumLink under filenames GEOWAR and GEOWAR.DOC, uploaded by Rob Siegel. On GEnie its file number is 6537, and on CompuServe its filename is GEOWAR.BIN. *(Screenshot taken from* geoWorld, *Issue #18, 1994, p. 11. –MS)*



This is a very popular GEOS game. That said, I have to admit that I never play it. Oh, don't get me wrong. It's an excellent rendition of the board game Risk—just different enough not to step on any copyright toes. It has great graphics and well-executed gameplay. It's just that I have never really had the patience or tactical know-how to do very well at Risk. Anyway, I like my war games filled with noise and explosions, as you know if you read my other column in this magazine.

Draw Poker (QuantumLink filename is DRAWPOKER.SDA, uploaded by Sir Gary. GEnie file number is 12449.) is another good time-waster. It's essentially mindless, just like the video poker machines it emulates, but who cares. I always like a gambling game that lets me get more money whenever I lose it all—which in my case is quite often. What's so cool about drawing a really good hand and collecting all those imaginary winnings? I don't know, but I do know that I'll keep *geoPoker* on my RAMLink, ready to interfere with my finishing another column. This one runs on the 128 in 40-column mode.

There are a lot more where that came from, believe me. Try *Cluster Wars,* for example, if you want to see superb graphics in a game, or *geoSlots* for some simple casino action. There are even arcade games around for GEOS, such as *Shoot Out* and *Breakout* on the *CMD PowerPack* disks. Paul Murdaugh, the fellow who created *DualTop,* also wrote a version of *Solitaire* that can compete with the snazzy version written for Windows. Hey, productivity is fine, but clearing a whole screen full of *geoTiles* really rounds out the work day, don't you think?

Machine Language

Jim Butterfield

BASIC Fun

Amaze your friends by tweaking your 64's BASIC! There's a clever trick that opens up new 64 possibilities.

Here's the story. Normally BASIC is in ROM (Read Only Memory) so that it cannot be changed. You can make the BASIC ROM disappear by storing a certain control value in address 1, and RAM will appear in its place. Before we do this, we can copy BASIC from ROM to RAM, making minor changes. We can do this copy at lightning speed using machine language.

Where BASIC Lives

BASIC lives at addresses 40960 to 49151. The addresses seem random. But change them to hexadecimal, and you'll start to see a pattern: from $A000 to $BFFF is exactly 8K of memory.

Our program's job is to take the contents of each of these ROM addresses, and copy the contents to the RAM beneath. That's easy, and it doesn't need any bank-switching, because of a feature called poke-through. We read data from ROM, but we store this data in RAM.

After we've done the copying, we can nudge the control value at address 1, and the ROM will appear to vanish—at least until we poke it back or turn off the power.

The Copying Job

Simple indexing can reach a range of only 256 bytes. To move 8K, we will use indirect indexed addressing, with the indirect address stored at hex $FD/FE. Let's set up that address to point to address $A000.

```
LDA #$A0 : LDY #$00 : STY $FD : STA $FE
```

Our indirect address is set, and register Y happens to contain a 0. Let's copy a byte from $A000 to $A000 (ROM to RAM, right?), using indirect addressing.

```
(lp) LDA ($FD),Y : STA ($FD),Y
```

We're ready to move along to the next address. The quickest way to do that is to bump the value in the Y register and then loop back to our LDA/STA pair above.

```
INY : BNE lp
```

If the branch failed, Y has gone "over the top" and is back to 0. Time to bump the high byte of the indirect address.

```
INC $FE
```

We'd better test that high byte. If it's reached a value of $C0, we are now beyond address $BFFF. In that case, we should quit. Otherwise, back to the loop.

```
LDX $FE : CPX #$C0 : BNE lp
```

There's a little more code in our program, but I'll leave that for you to explore. It includes placing the new value into address 1 so as to make the ROM disappear.

The BASIC Program

Load the following program into your 64. It's listed on the disk as XBASIC.ML. It will poke the machine language program into place and execute it. Here's the listing for you to examine.

```
90 REM: PROGRAM FOR COMMODORE 64 ONLY
100 DATA 169,160,160,0,132,253,133,254
110 DATA 177,253,145,253,200,208,249
120 DATA 230,254,166,254,224,192, 208,241
130 DATA 169,85,141,162,160,141,251,160
140 DATA 169,65,141,5,161,141,14,161
150 DATA 169,54,133,1,96
200 FOR J=828 TO 871
210 READ X:T=T+X
220 POKE J,X
230 NEXT J
240 IF T<>7094 THEN STOP
250 SYS 828
270 PRINT "BASIC IS NOW IN RAM!"
280 PRINT "(WITH MINOR CHANGES)"
290 PRINT "POKE 1,55 TO RESTORE!"
```

Run the program, and your system will be reconfigured.

Now for the Fun

BASIC in ROM, BASIC in RAM—what's the difference? It's still BASIC, isn't it?

Yes it is, and it will still load and run virtually any BASIC program you have. But there's something new. ROM cannot be changed, but you can change RAM.

For your amusement and education, I've slipped in four changes to BASIC keywords.

Type LIST to display the program, and you'll discover that it doesn't work! I've sneakily changed the letter I to an A; you'll have to command LAST to do the job.

When you do get the program listed, look at it carefully. Three keywords are no longer what they used to be. They still work; only their appearance has changed.

The modified keywords display differently. You must type them in their new form, or you'll get a SYNTAX ERROR message. You know about LIST, but can you find the others?

Background

The 64 is fascinating. You can kick out not only BASIC but also the Kernal ROM and the I/O chips. All this is done by modifying the contents of address 1. It must be done carefully, of course, since you may lose control of your 64 if some of the logic is missing. Everything returns to normal, however, when the power is switched off.

Serious machine language programs often kick out BASIC to gain access to more RAM. Some utility programs arrange to hide underneath ROM, where they are less likely to come into conflict with other programs.

More Amusements

Feel free to snoop around within BASIC. A machine language monitor is handy for this, although you can chop away with just BASIC. Don't worry about damaging anything. Anything you do can be fixed by turning off the power.

When you've finished, you can return to normal BASIC. The program tells you that commanding POKE 1,55 will slide the ROM back into place.

That's almost true. But I've made a slight change to the keyword POKE. Can you figure out what you really have to type to restore the system?

Beginner BASIC

Larry Cotton

Lessons

During a recent marathon programming session for a new game called Setz! (which you'll find on this disk), I kept a rough diary of my programming efforts. This record contained especially the lessons that I learned (and re-learned) as I went. I documented enough errors, careless mistakes, and obvious glitches to serve as fodder for this column for several months.

Not to worry; I won't bore you with every mind-numbing bungle. Let's just look at several high (or low) spots.

Losing Track of Variables

When I begin a programming session, I just start typing. I'm in too much of a hurry to be a structurist. BASIC (especially Commodore's version) doesn't care if there's a GOTO on every other line. I know roughly what I'm trying to accomplish, but I don't know how long the program will be, how many subroutines it will take, or how many variables I'll need.

Before long I'm wondering, "Have I used this variable before?" or "What meaningful name do I give this variable?" As you may know (or recall from previous columns), variable names can be any practicable length, but the 64 and 128 lop off everything but the first two letters.

Thus you can say AREA = 4 or COLOR$ = "RED", but the computer only remembers AR = 4 and CO$ = "RED". Later, if you use the variable ARLO or COPY$, the computer won't be able to distinguish between AREA and ARLO or between COLOR$ and COPY$.

It's usually good practice to recycle one-letter variable names, such as I and J, in FOR-NEXT loops. If you do this, be sure you don't use I or J for anything else: a FOR-NEXT loop may have changed one of their values.

Try to use meaningful variable names, such as LE for length or WI for width. However, if you're timing something, don't try to use the variable TI, which is reserved for the computer's exclusive use. Likewise, you can't say STATUS = 24, because ST is also reserved. Use of BASIC keywords in variable names is also verboten. You can't say ONYX = 14, TOTAL = 56, or OREGON = 41 because those words start with ON, TO, and OR, respectively. I recall a classic no-no of a few years ago: MANDY = 12. Wrong. The computer interprets this as M AND Y = 12—a blatant syntax error.

Finally, keep a list of variables on a piece of paper as you go. On your third all-night programming session, this list will be invaluable.

Too Many GOTOs

I guess if there's one mistake I can't seem to avoid, it's that I use too many GOTOs. (Some BASIC programmers would say one GOTO is too many.) Try to plan far enough ahead to use GOSUBs instead. Put your often-used subroutines at the end of the program. It's cleaner, it's easier to follow by someone trying to figure out the program, and it can conserve memory.

My problem, as I said, is that I just start typing. In SETZ!, I began with the instructions right up front (that's the first thing the user sees), but later I was forced to rewrite the instructions as a subroutine, which I moved to the end of the program for clarity. Also write setup routines (screen clearing, variable defining, ML poking, and so on) at the end of the program (using high line numbers) and call these subroutines as needed.

If you find yourself writing the same line over and over with only minor variations, try making variables out of the variations. This line may appear early.

```
100 POKE53280,12:POKE53281,6
```

And this variation may appear later.

```
500 POKE53280,13:POKE53281,5
```

Instead, define 53280 early in the program, change lines 100 and 500, and write a short subroutine with a high line number. Here's an example.

```
10 V=53280
```

(other stuff)

```
100 CB=12:CS=6:GOSUB1000
500 CB=13:CS=5:GOSUB1000
```

(other stuff)

```
1000 POKEV,CB:POKEV+1,CS:RETURN
```

Too Many IF-THENs

IF-THENs can be used to look at keyboard responses.

```
100 GETA$:IFA$=""THEN100
110 IFA$="1"THENGOSUB1000
120 IFA$="2"THENGOSUB2000
130 IFA$="3"THENGOSUB3000
```

A more sophisticated approach uses ON-GOSUB and VAL.

```
100 GETA$:IFA$=""THEN100
110 ONVAL(A$)GOSUB1000,2000,3000
```

IF-THENs can be used to control program flow based on a variable's value.

```
100 IFX=45THENGOSUB1000
110 IFX=55THENGOSUB2000
120 IFX=65THENGOSUB3000
```

Better:

```
100 ON(X-45)/10+1 GOSUB1000,2000,3000
```

In SETZ! I wanted to turn on the first sprite if U had a value of 0, the first two sprites if U equaled 1, and the first three sprites if U was 2. I could have written the following lines.

```
100 IFU=0THENPOKEV+21,1
110 IFU=1THENPOKEV+21,3
120 IFU=2THENPOKEV+21,7
```

Instead, I programmed it this way.

```
100 POKEV+21,U^2+U+1
```

Thus as U changes from 0 to 1 to 2, the values that get poked to V+21 will be 1, 3, and 7. It takes a bit of head-scratching to save a few lines of code, but you'll feel better about your program's efficiency—and you'll enjoy the mental stimulation!

Too Many PRINT:PRINT:PRINTs

How do you print a message in the middle of the screen? Here's one way.

```
10 FORI=1TO10:PRINT:NEXT
20 PRINT"[12 SPACES]MIDDLE OF SCREEN"
```

Or, worse, you could use numerous arcane cursor-positioning symbols.

```
10 PRINT"[DOWN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN]
[RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT]
[RIGHT][RIGHT]MIDDLE OF SCREEN"
```

A better way would be to write a neat little subroutine using two valuable POKEable memory registers: 214 (vertical position of cursor) and 211 (horizontal position of cursor). Here's how that might work.

```
1000 POKE214,Y: PRINT: POKE211,X: PRINTM$: RETURN
```

Be sure to put PRINT between the two pokes. Then, when you want to print a midscreen message, just enter something like the following.

```
10 Y=10:X=12:M$="MIDDLE OF SCREEN"
20 GOSUB1000
```

You could even count the number of characters in the message, subtract from 40 (the screen's width in columns), and divide by 2.

```
10 Y=10:M$="MIDDLE OF SCREEN"
20 GOSUB1000
```

The subroutine would be as follows.

```
1000 POKE214,Y: PRINT: POKE211,((40-LEN(M$))/2): PRINTM$: RETURN
```

Next month we'll look at a few more common programming errors and tricks to help you avoid them.

PD Picks

Steve Vander Ark

Pegasis and Ghoul Dogs

What a great job. Every month, as the deadline for this column gets near, I spend most of my free time for a week or so prowling the libraries on QuantumLink and GEnie, playing with all kinds of programs that I turn up. Many of these programs are games. When my wife frowns and says something about wasting all my time blowing up aliens or playing computer card games, I get to honestly tell her that I'm working. Like I said, this is a great job.

Eventually, though, I have to choose which programs I will include in this column and stop playing and start writing. It's usually pretty easy to choose a few really great programs; it's not always so easy to get to work (the kind my wife identifies as work, that is). But now here I sit,

with my joystick nowhere to be seen, working my little heart out. I can see my wife smile with satisfaction.

So here are this month's goodies. One of them is an oldie but goldie that you can find on QuantumLink and CompuServe, but the other is available only on GEnie. Quite honestly, I found quite a few more where these came from, and if disk space wasn't a limiting factor, I could hit you with a list that would knock your socks off. As it is, I'll have to spread them over a few months. Bummer. Guess I'll have to spend a few more months with joystick in hand and my trusty can of Mountain Dew by my side. My wife will be thrilled.

Pegasis
Q-Link file names: PEGASIS uploaded by ISIL, plus PG.OBJA and PG.OBJB uploaded by Papa Doc. (You need all three of these files. Load and run PEGASIS with the ,8,1 extension.)
CompuServe filename: PEGASI.BIN.
GEnie file number: 13829.

A lot of advances have been made in Commodore programming since this game was uploaded to QuantumLink in 1986, but you'd hardly know it. *Pegasis* is a beautiful game—even if the mythical flying horse is usually called Pegasus. The background scenery, although more or less unimportant to gameplay, is spectacular. It makes some of the best use of color I've seen in a public domain game. The sprites are finely detailed and artistically done. Even the lettering of the GAME OVER message (which I saw far too often) is elegant and beautiful.

I have to be honest here. Nothing at all blows up in *Pegasis.* The game, which resembles the arcade game *Joust,* requires you to fly a winged steed and duel in the air with assorted nasties. All you have to do is bop them from above to make them drop to the ground. You don't fire anything at them; you don't swing a sword. Oh, they do come at you faster and faster as the levels roll by, and it's gratifying when you manage to do a triple bump. But you know how I am about flying debris and mind-boggling weaponry. This game has none of that.

What it does have is style. The graphics, which I mentioned above, are, well, elegant. So is the way your flying horse changes into a pure white, riderless image before dissolving in a scatter of flecks when it gets hit. The physics of the game—the way you use the joystick button to keep your horse flying and the feel of gravity that influences your movements—is intriguing enough to keep you playing but not so weird as to frustrate you. Even the sound effects are subtly effective.

All in all, *Pegasis* is the kind of game that is a pleasure to play. It even includes a scrolling screen of text giving some background on the mythology upon which the game is based. Even if you prefer your games wild and woolly and filled with things that explode when you shoot them, you'll still appreciate a game with a flair for the artistic and elegant. In other words, you'll like *Pegasis.*

Ghoul Dogs
GEnie file number: 13724.

This game doesn't have anything blowing up in it and isn't elegant. I know, I know, you wonder what in the heck I'm thinking. Aside from a fairly exciting name, what can *Ghoul Dogs* offer?

I'm not entirely sure why I like this game. Maybe it's because the first time I started it up I laughed out loud. Maybe it's the Tony the Tiger look-alike with the smug grin that you push around the screen. Maybe it's those ferocious dog faces with their colorful eyes that flit around after you. I don't know. But I've played this game over and over, and I love it.

I suppose one reason might be the fact that my cat can get chomped plenty of times by those dogs before the game finally ends. For a guy like me, whose five-year-old daughter beats him at Nintendo games, it's nice to have a game that lets me keep going for awhile. As the levels increase, the dogs get taller and wider and start moving faster. Even when I scarcely make it out of the corner before one of those beasts strike, I get resurrected. Of course, the game eventually finishes, but not before I get a lot of tries. I like that.

If scooting around out of the reach of four or five bad guys was all this game had to offer, it wouldn't be much of a game. But *Ghoul Dogs* tosses in an added attraction: You get to shoot targets rolling around the outside of the playing field. OK, you're probably thinking that's why Vander Ark likes it—it has shooting in it! But in this case there are no burly sound effects and absolutely nothing blowing into pieces. As a matter of fact, all you get is a rather boring beep when you hit something. That's it, but it's fun anyway. The graphics are a kick, the game keeps hopping along with a few variations here and there—and it has shooting.

Before I wander off, I have to tell you about a new program on CompuServe called *C64EMU*. It's a Commodore 64 emulator for the IBM compatible, complete with built-in Commodore BASIC and a couple of pseudo-disk drives. With a Sound Blaster sound card you can even get simulated SID sounds. Unfortunately, it runs a little too slowly on my PC to be of much use, but it's interesting to see that familiar blue startup screen on my VGA monitor! If you belong to CompuServe and want to check this one out, it's in the Commodore Applications forum.

D'Iversions

Fred D'Ignazio

The Real MPCs: Kids Inventing Multimedia on Their Own

A teacher called me two nights ago and asked about the new MPC (Multimedia Personal Computers) which supposedly represent the next generation of computing. "Why should

schools buy them?" the teacher asked. "Why are they so much better than the 286 and 386 computers that schools already have?"

"What 286 and 386 computers?" I asked, thinking about all the schools I visit that lack high-powered computers but still have PCjrs, Commodore VICs and 64s, Radio Shack TRS-80s, Atari 400s, TI 99/4As, and older Apple IIs.

"You know what I mean," she said. "Older machines. Are MPCs any better than what teachers already have?"

Then I gave her an answer which surprised her. I said that the new MPC computers really do show promise, and I recommended that schools invest in one or two of them to get their feet wet with new MPC technologies like CD-ROM encyclopedias.

However, these hot new machines are not the dawn of a new era but its demise. They are a signal that personal computers are finally on their way out. Like tail fins on Cadillacs, today's desktop computers are the gaudy last hurrah of personal computing. And just as the era of the personal computer fades into oblivion, a new era is being born—the era of personal TV.

The truth is harsh. MPC machines today are ramshackle gyro gearloose contraptions with cables, wires, cards, and boxes that require their own corner of the living room just to accommodate all the gear. It's true that legions of MPC hobbyists take these devices home and fuss with them as if they were the model trains of yesteryear or high-tech erector sets. But for most people, today's MPCs are not the friendly, easy-to-use media appliances that their manufacturers claim them to be. Instead, they are unruly creatures—pets—that take up a lot more time than they're worth.

And like many other endangered creatures, the MPC's time is running out.

Now, let's think about the next five years. Next-generation computers will be completely different. They will be multimedia wizards, but they will be buried in a little black box resting atop a TV set and be the "understudy" to the new star: the interactive television.

The TV screen will be transformed into a two-way "tele-mall" connected to an infinity of delicious consumer products and experiences via the computer, cable TV, and the household telephone.

How will we shop at this new tele-mall? One thing's certain: We won't sit up close to the computer the way we do today. The keyboard and the standard desktop on the computer screen will disappear and be replaced by what consumers who are surveyed say they really want: flat-screen LCD "zappers" that instantly call up menu buttons on the TV set.

These buttons, with bold, brightly colored Sesame Street letters and cartoons, will let the interactive user do virtual shopping at Macy's; play movie-like video games; interact with national quiz shows and soap operas; attend classrooms without walls and virtual-reality family reunions; and then take exotic swear-you-are-there vacations to go undersea treasure hunting, lunar skydiving, or dirt biking through Maya ruins.

To fully immerse themselves in these virtual shop-a-thons and adventures, the family members will be able to choose from a diverse selection of TV screen sizes. Screens will range

from giant, flat, wall-sized home theater screens to tiny stereoscopic goggles and "stun glasses" that allow the user to take a solo spin on an electronic highway without the hassle of having to share the experience with others in the same room.

Today's primitive MPC video clips, music, and spoken voices will be replaced by Hollywood-quality images and sounds. Today's obscure, boring databases will be enclosed in "media base" shells which clothe symbolic information (numbers, letters, words, formulas, diagrams, and so on) in rich, vivid real-world experiences.

For example, if you were looking for the name of a nearby garage to service your car, you wouldn't go to the yellow pages. Instead you'd run into your family den, turn on the smart TV, and take off in an aerial flyby of your local community. As you soared along in a two-seater airplane, you would look down and see cartoon mechanics waving at you from their store parking lots. "Come visit us," they would say. "Lowest-priced auto service in town!"

Or maybe you were in the market for a new apartment or home. Your search of local homes on the market in the real estate ads would be replaced by a virtual tour of the homes in which you could zoom in and take a virtual tour of each room while performing a "picture-in-picture" comparison of the other homes you had visited on your tour.

Similarly, you might be browsing the movie section of your "video newspaper," point and click on a new movie, and see a 30-second preview of the movie explode onto your TV set.

TV shows, diamond rings, L.L. Bean hiking boots, a high school course in algebra, sitcoms, and fast-breaking global news events would all be data—all part of the TV's online pay-per-view files available to any consumer with instant credit.

So What Should Schools Do?

"If computers are going to disappear into television sets, then maybe schools shouldn't be buying MPC computers," said the teacher. "Maybe I should tell them to just wait."

"No, probably not," I answered.

"Then what can they do instead?" she asked.

That's when I told her about an alternative. "Teachers don't have to wait for computers to merge and disappear inside other machines," I said. "Right this minute, teachers can enlist their students and experiment with this same process of fusion and transformation in the classroom."

"But won't this mean buying a lot of new equipment?" the teacher asked.

"No," I said. "That would be the old approach. Students and teachers can scavenge almost everything they need right in their school building—and what they lack is probably in students' bedrooms and can be shared with the classroom on a semipermanent loan.

"Students and teachers can use their humble little classroom computer as a hub of a smart TV by putting the computer on a cart and adding to it things like an old record player, a tape recorder or boom box, a video camera, and a VCR and TV."

"I don't understand," said the teacher. "What's the purpose of linking all these old boxes to our computer?"

"Because your students and teachers would be building the future," I said. "Students and teachers can reconfigure computers and other low-tech devices into a multimedia inquiry center for the classroom. It'd be affordable and is far more educational than the TV-computer-telephone MPC hybrids of the future.

"Just letting kids act with some smart TV in the future," I continued, "doesn't make for an educationally complete experience. Kids using the multimedia TV sets will seem like they are doing totally new futuristic learning, but they will still be couch potatoes—hyper couch potatoes. And they'll be missing some essential experiences along the way."

"Like what?" asked the teacher.

"Like making up experiences on their own. Like authoring, and investigating, and problem solving, and extending their learning into their own community, and connecting the virtual worlds they're being fed on the TV screen with the real world just outside the classroom door. And like working together so that moments of learning occur with other people and not just with the computer.

"The most important reason for students to invent their own MPCs and their own smart TVs is to promote the real basics in classrooms. The basics have nothing to do with French or pronouns or geometry. The basics are collaboration, problem solving, invention, and innovation. Students can experience all these elements of learning if we let them become co-creators in inventing the future. But if we just bide our time until a textbook company or Paramount Pictures or Steven Spielberg comes up with the future, then this opportunity will be taken away forever."

Programmer's Page

Dave Pankhurst

SpeedScript Programming

In January 1984, *Gazette* readers had a new program to enter and run. It was a word processor called *SpeedScript.* Now, ten years later, it's probably the most popular word processor in 64-dom and undoubtedly the most widely known.

I believe its popularity is in part due to the open nature of the program. From the beginning, people were patching and poking to change the way it worked. There was even a book with the source code, allowing programmers to tinker with its internals. This tinkering resulted in new commands, useful support programs, and a wealth of practical information. This month's column taps into that wealth, showing that *SpeedScript* can still hold a few surprises, even in its second decade.

Customizing SpeedScript

These pokes let you customize *SpeedScript* to your own liking. Just load *SpeedScript,* but do not run it. Enter the changes in immediate mode, and then save the altered *SpeedScript* version to disk. Be sure to use a descriptive name to avoid any mix-ups.

LOCATION	CONTENTS
5722,5723	Left, right margins
5724	Page length
5725,5726	Top, bottom margin
5727	Line spacing
3372	Text color
3357	Background color
4288	CLEAR command color
3858	DELETE command color
4556	ERASE command color
2596	EXIT command color
4258	CTRL I command color
5085,5104	LOAD command color

The command colors refer to the colors of the top line on the screen when *SpeedScript* performs these different commands. For instance, POKE 4288,15 would set the top line during memory clearing to light gray.

SpeedScript can be accessed and improved with machine language routines or patches. One such patch lets you skip the annoying TAPE OR DISK message. Here are the POKEs to enter after loading *SpeedScript.*

```
POKE 4904,44:POKE 4909,224:POKE 4910,8
```

Other, larger routines need more room, and for that, memory from $C000 to $CFFF is ideal. The most significant byte of the pointer to the top of *SpeedScript's* memory is stored at 2481; poking 191 to it will protect from $C000 on up. A location for smaller patches is from 9329 to 9472, but be warned that others have thought of it before you. (*WordCount* is one such program.) Placing a routine there lets you load and save it along with *SpeedScript,* instead of using a cumbersome boot program.

Speaking of cumbersome boot programs, here are several that you may find useful. Note that these all load and patch *SpeedScript* in memory. They don't change the program on disk, so your original program remains unaltered. The disadvantage to these programs is that you have to run them every time you use *SpeedScript.*

Load and save these boot prgrams to a work disk that contains a copy of *SpeedScript.* Since these programs look for and attempt to load the filename SPEEDSCRIPT, either change the name in the boot program to whatever you've renamed *SpeedScript* or be sure that SPEEDSCRIPT is the filename on your disk.

Larger Character Buffer

The first program should be a boon to fast typists. I prefer to type in insert mode, which means that on larger files, the computer can be so busy inserting that it ignores some of my keystrokes. Being only 10 characters long, the Commodore buffer quickly fills up. This program increases the buffer to 123 characters; to date, I haven't been able to overflow it.

Here's the listing for informational purposes. The ready-to-run program is on th other side of this disk. It's called SSBUFFER XTND.PP. Copy this program and save it to a work disk that contains *SpeedScript*.

```
10 REM SPEEDSCRIPT BUFFER EXTEND PROG
20 PRINT"[CLR][DOWN][DOWN]L[SHIFT O]" CHR$(34)"SPEEDSCRIPT*"
CHR$(34)",8,1"
25 PRINT"[7 DOWN]POKE2481,197: SYS52992: RUN[HOME]";
30 FORI=631TO640: POKEI,13: NEXT: POKE198,10: END
35 DATA120,169,29,141,42,3, 169,207,141,43,3,169,0,141,
255,207,169,66,141
40 DATA143,2,169,207,141,144, 2,88,96,124,165,153,240,3,76,
74,241,172,255
45 DATA207,240,23,120,172,128,207, 162,0,189,129,207,157,128,
207,232,236,255
50 DATA207,208,244,206,255,207, 152,88,24,96,32,72,235,165,
198,240,22,169
55 DATA0,133,198,173,119,2,174, 255,207,236,28,207,176,7,157,
128,207,232,142
60 DATA255,207,96,999
```

A word of caution: Don't hold down repeating keys. They fill the buffer quickly. The result is long waits, as you watch all those keys being executed.

Passwords for Files

On the cloak-and-dagger side, this program lets you load and save *SpeedScript* files using a secret code. Once again, here's the listing for you to examine. The program itself, SSFILECODER.PP, is on the flip side of this disk.

```
100 REM SPEEDSCRIPT FILE ENCODER
105 X=51000
110 READX$: PRINT".";: IFX$=""THEN130
115 FORI=1TOLEN(X$)STEP2: Z$=MID$(X$,I,1): Y=ASC(Z$)-48+7*(Z$>"@")
120 Z$=MID$(X$,I+1,1): Y=16*Y+ASC(Z$)-48+7*(Z$>"@")
125 POKEX,Y:X=X+1: NEXT: GOTO110
130 PRINT"[CLR][DOWN][DOWN]L[SHIFT O]" CHR$(34)"SPEEDSCRIPT*"
CHR$(34)",8,1"
135 PRINT"[7 DOWN]SYS51000: POKE2481,197: RUN[HOME]";
140 FORI=631TO640: POKEI,13: NEXT:POKE198,10:END
145 DATA A9C78D3303A94D8D3203A9C78D3103A9898D3003
150 DATA 60A5028DDBC8A5038DDCC8200CC8D0034CEDF5A5
155 DATA C18DE1C8A5C28DE2C8A6AEA4AF20CBC720EDF508
160 DATA AEE3C8ACE4C820CBC7ADDBC88502ADDCC8850328
165 DATA 6048C900F004684CA5F4A5028DDBC8A5038DDCC8
170 DATA A5C38DE1C8A5C48DE2C8200CC8F0E36820A5F4B0
175 DATA 1908488A48984820CBC7ADDBC88502ADDCC88503
180 DATA 68A868AA6828608EE3C88CE4C8A000A200ADE1C8
```

```
185 DATA 8502ADE2C88503AEDAC8E8CAD003AEDAC838A502
190 DATA EDE3C8A503EDE4C8B015B1025DE5C8910218A901
195 DATA 650285029002E6031890D860A278BDB7059D07C9
200 DATA BD61C89DB705CAD0F1A9008DDAC8A5C6F0FC7820
205 DATA B4E5C90DF025C914D00FCEDAC830E6AEDAC8A920
210 DATA 9DF105D0E1EEDAC8AEDAC89DE5C8A9189DF005E0
215 DATA 1590CFA278BD07C99DB705CAD0F7ADDAC8607040
220 DATA 40404040404040404040404040404040404040
225 DATA 40404040404040404040404040404040406E5D05
230 DATA 0E14051220100113131 70F1204203E2020202020
235 DATA 20202020202020202020202020202020203C5D6D40
240 DATA 40404040404040404040404040404040404040
245 DATA 40404040404040404040404040404040407D00,
```

After it's up and running, all SAVEs and LOADs go through this routine. You're asked for an optional password. To save or load normally, just press Return. When you enter a password, the file is encoded and saved. To load the file later, you must re-enter the password exactly as you entered it the first time. If you don't, the file will look like garbage.

The program works by exclusive-ORing your code with the file. So while it's reasonably secure for personal data, don't plan to store state secrets with it.

Drive Swap

This last program allows *SpeedScript* to work with two disk drives. Pressing Ctrl-N (for New drive) toggles between devices 8 and 9, displaying the current device number at the top of the screen. SAVEs, LOADs, and directory listings all work with this new device number. Pressing Ctrl-N again reverses it. Be sure to have both devices up and running, or *SpeedScript* could hang up. This program is on th flip side and is called SSDEVICE SWAP.PP.

```
10 REM SpeedScript DEVICE SWAP
20 FORI=52736TOI+55: READX: POKEI,X:Z=Z+X: NEXT: IFZ=5475 THEN
PRINT"DATA ERROR":STOP
30 PRINT"[CLR][DOWN][DOWN]]L[SHIFT-O]" CHR$(34)"SPEEDSCRIPT*"
CHR$(34)",8,1"
40 PRINT"[7 DOWN]POKE2481,197: POKE2854,0: POKE2855,206: RUN[HOME]";
AQ 50 FORI=631TO640: POKEI,13: NEXT: POKE198,10: END
60 DATA44,8,9,201,14,208,46,32,78, 10,174,1,206,236,235,18,208,3,
174,2,206
70 DATA142,235,18,142,44,19,142,154, 20,142,241,22,142,79,23,142,
223,24,142
80 DATA227,26,142,91,27,142,161, 27,169,0,32,1,30,76,105,10
```

If you have a programming tip or hint that you'd like to share with our readers, send it on disk to:

Programmer's Page
COMPUTE's Gazette
324 West Wendover Avenue, Suite 200
Greensboro, North Carolina 27408

We pay $25-$50 for each tip that we publish.

Review

Don Radler

Full Count Baseball

Game four of the 1993 World Series was the highest-scoring game in Series history. The blowout came the night before I was asked to review another baseball simulation, this one known as *Full Count Baseball.* Sure, I said, and then headed to the computer to prepare myself to test the new piece of software.

For purposes of comparison, I noodled around with a couple of the baseball programs I had on hand—*Hardball* and *Street Sports Baseball.* With neither of these could I hit as well as either team in game four of the series—imagine 29 runs in one series game!—and with neither could I pitch much better than the surprisingly ineffective hurling on both sides in that game.

Hardball presented some managerial choices, but nothing to match the dilemma Cito Gaston faced in deploying league batting leaders Molitor and Olerud in Philadelphia without his accustomed designated hitter option.

The Street Sports kids were cute, but not nearly as interesting as the scruffy maulers from Philadelphia. The Phillies' attitude was summed up in a single quote: Right before game six, facing the return to Toronto's SkyDome and what the sports writers all called "sudden death," the Phillies' John Kruk was heard to say, "If you're gonna die, it might as well be sudden."

There have been many baseball simulations since the earliest Commodore days. Some were largely statistical, such as *Computer Baseball;* others were almost pure arcade games, such as *Star League Baseball.* Most fall somewhere in between, letting you do more than just wiggle your joystick to pitch and hit and giving you a little of the feel of the game along with some of its stats.

Lance Haffner Games produced *Full Count Baseball,* and Haffner declares it "the finest baseball simulation on the market for any machine." Statistically, it's an impressive piece of programming, but any simulation should give you the look and the feel of the activity it simulates. In *Full Count Baseball,* there are none of the sights, none of the sounds, none of the personality of the game of baseball. Indeed, there is nothing but strategy and statistics. Lots of statistics.

Although the manual talks about "playing" a game, this is really a manager's program. Given certain players in certain positions, with a particular lineup for each side, how would the game come out? The playing options along the way—hit & run, bunt, steal—seem to be less significant than who is performing the action.

Events are described laconically on a bare-bones text screen with nary a bat or a ball in sight and without any of the organ music, hollering, crack of the bat or other sounds that give a ball game its atmosphere.

You can play *Full Count Baseball* against a friend, with the computer acting as umpire and scorekeeper. You can play against the computer. Or you can let the computer handle both

teams and watch the onscreen text change as computer-generated hits, runs and errors take place.

On its game disk, *Full Count Baseball* also gives you 26 current American League and National League team rosters. On another disk, there are the rosters of 52 past teams. All rosters include batting, fielding, and pitching statistics. For batters, there are 21 categories of data, including whether they bat lefthanded, righthanded, or both; principal positions played; total at bats, hits, doubles, triples, and homers; RBIs; strikeouts; steals; number of times caught stealing; running and fielding ratings; and so on.

For pitchers, there are almost as many categories, including games won and lost, games started, games saved, hits allowed, strikeouts, earned run average, home runs allowed, and so forth. All of this can be viewed on screen or printed out; this is true of all compilations of data with this program.

After you've loaded the game and a couple of team rosters, you pick a starting pitcher and, optionally, a complete pitching strategy. If the computer is playing, you have to make this choice for it, too. Then you work out a lineup for your team and the computer's team(s). This initial setup before a game takes a good ten minutes once you have learned how to do it; it took a lot longer than that for me the first several times I tried it.

For a player-controlled team in the field, there are seven defensive options, including pitch, intentional walk, infield in, first and third basemen in, or infield normal, and personnel change, including bringing in a relief pitcher. The player-controlled team at bat has five offensive options, including hit, hit & run, bunt, steal and bring in a pinch hitter.

When the computer manages a team, it exercises the above options according to ratings based on statistical probability. (When it does this against any team I'm managing, the computer seems to beat me every time.)

There is no joystick action with Full Count Baseball. To choose to hit away or, when in the field, to walk a batter, you type in the number of your selection. All events that then take place at bat and in the field are announced on a black-and-white text screen. The printed lines are reminiscent of radio play-by-play coverage, "Alomar is going...SAFE AT SECOND!" and so on. Game results can be sent to the printer and/or saved on disk, and team stats and ratings can be recompiled based on each new game's results.

You can form your own teams, drafting players from any of the teams on disk or creating new teams with stats to be supplied by you. You can also trade players from one team to another. You can also form your own league, competing with other baseball fans who have this software. A third disk bundled with the game provides a league standings program.

If you're the kind of true baseball fan who always keeps score and who loves to second-guess every call any manager makes, *Full Count Baseball* may be just your bag. There are no bells and whistles, but there is a wealth of data on real-world teams (ten additional multiteam disks are available at $19.95 each) and there are all kinds of ways to use this information, or even to change it if you wish.

For what it's worth, just before the sixth and final game of the 1993 series, I pulled up the most recent rosters of the Phillies and the Blue Jays, and then had the computer play both

sides with lineups as close as possible to those used in game six. (The real Toronto team had changed about half its roster since Haffner's Current Teams disk was prepared, of course, and there had been some Philadelphia changes as well.) The outcome: Toronto 6, Philadelphia 3. If you were a betting man, this statistical wonder might do something for you: in the real world, the score was pretty close to the computer's prediction: 8-6 Toronto.

The Phillies were leading 6-5 in the middle of the ninth, but sudden death struck them when Toronto's Joe Carter hit a three-run homer in the bottom of the final inning. All this happened right in front of all those Canadian fans, and Toronto had won its second series in a row. As you might expect, pandemonium ensued.

Or, as *Full Count Baseball* might have put it, "Carter homers. Blue Jays win."

LANCE HAFFNER GAMES
P.O. Box 100594
Nashville, TN 37224
(615)242-2617
$39.95; S&H $3.00

Commodore Buyer's Guide

Compiled by Lisa Hayes

Finding products and services for your 64 or 128 might be a little easier with this list of firms that still support the 8-bit Commodore line.

Naturally, some of these companies may have closed since this list was compiled, but it should give you a good start for finding items you need for your Commodore. To make any additions, changes, or deletions to this list, please contact Tom Netsel, COMPUTE's Gazette, 324 West Wendover Avenue, Suite 200, Greensboro, North Carolina 27408.

A & M Computer Repair
24 Colonel Conklin Dr.
Stoney Point, NY 10980
(800) 344-4102
(914) 947-3522
Authorized Commodore repair center that also sells parts and power supplies.

AGT Software
Leesburg, GA 31763-0938

Antigrav Tool Kit
P.O. Box 1074
Cambridge, MA 02142
Offers GEOS software. Also makes the Key DOS ROM chip version 2 for the 128/D.

Andor House
3907 S.E. 2nd Ave.
Cape Coral, FL 33904
Low-cost music and computer educational software.

Bare Bones Software
940 4th Ave., Ste. #222
Huntington, WV 25701
(800) 638-1123 (orders)
(304) 529-0461 (customer service)
(304) 529-7050 (BBS)
Buys and sells used software. Free catalog. (Specify 64/128, Amiga, or IBM.)

Basic Bits Commodore Group
P.O. Box 39447
N. Ridgeville, OH 44039-0447
(216) 243-0887
This user group offers out-of-town members an associate membership for $10. Members receive a monthly newsletter and access to the group's PD library as well as computer support.

Bear Technologies
1005 McKean St.
Philadelphia PA 19148
(800) 755-5295
(215) 336-5296
Commodore repair service.

Brantford Educational Services
P.O. Box 1327
222 Portage Rd.
Lewiston, NY 14092
(519) 756-6534
Offers self-tutoring basic lessons for the 64, 128, and Amiga.

Brown Boxes, Inc.
26 Concord Rd.
Bedford, MA 01730
(617) 275-0090 or 812-3675
Specialized hardware: battery-backed cartridges.

Busy Bee Software
P.O. Box 2959
Lompoc, CA 93438
(805) 736-8184
Home of *The Write Stuff,* word processors for the 64 and 128. Many support programs for TWS are also available: *BB Talker, The Illustrator I, BB Templates,* and so on.

Busy Bee User Group
P.O. Box 3
Arivaca, AZ 85601-0003
A one-year membership is $20. This includes a subscription to the group's informative newsletter.

Cabral's Software
Box 98
W. Wareham, MA 02576
PD and CP/M software. Some new and used commercial software. Send 50 cents for a CP/M catalog on disk.

Caloke Industries
P.O. Box 18477
Raytown, MO 64133
(816) 478-6185
PD and shareware software. Free catalog.

Cee 64 Alive!
P.O. Box 232115
Sacramento, CA 95823
(916) 395-9773
Bimonthly disk publication for the 64. One-year subscription is $25.

Centsible Software
P.O. Box 930
St. Joseph, MI 49085
(616) 428-9096
(616) 429-7211 (BBS)
Used software. Very extensive listing. Free catalog. Specify 64/128, Amiga, or IBM.

Clip Art Cupboard
P.O. Box 317774
Cincinnati, OH 45231
Graphics for desktop publishing.

Codeware
Box 3091
Nashua, NH 03061
(603) 891-1588
Specialized goodies: a Morse code and RTTY program for the 64 and VIC-20, a 64 and VIC-20 tone decoder kit, a RTTY interface unit, and an EPROM programmer. Send $1 for catalog.

Commodore Country
Rt. 1, Box 333
Burleson, TX 76028
(817) 295-7658
Buys, sells, and trades used hardware and software. Also does repairs and sells chips and accessories. Free catalog.

Commodore Heaven
690 Primrose Dr.
Central Point, OR 97502
New software. Carries hard-to-find and out-of-print software titles. Send a business size self-addressed stamped envelope for catalog.

CNCN COM-NET
P.O. Box 390
Woodsfield, OH 43793
COM-NET Newsletter is an impressive bimonthly newsletter published by the Commodore Network User Group (a by-mail user group). Subscription rate is $12 for one year (six issues).

Compsult
P.O. Box 5160
San Luis Obispo, CA 93403
(805) 544-6616 (inquiries)
Commercial software closeouts. Catalog is $2 or free with any order.

CompuServe Info Service
P.O. Box 18161
Columbus, OH 43272-4630
(800) 848-8199
Nationwide computer online service.

Compustuff
P.O. Box 58317
Medina, OH 44258
(216) 725-7729
PD and shareware software. Also ham radio utilities.

COMPUTE'S Gazette
P.O. Box 3244
Harlan, IA 51593-2424
(800) 727-6937
Get 12 issues of the new *Gazette Disk* for $29.95.

Computer Bargain Store
3366 S. 2300 E
Salt Lake City, UT 84109
(801) 466-8084
Carries hardware and software. Some hard-to-find items. There's also a small inventory for the VIC-20 and Plus 4. Send $1 for a catalog.

Computer Workshops
3612 Birdie Dr.
La Mesa, CA 91941

Cosmi
2600 Homestead Pl.
Rancho Dominguez, CA 90220-5610
Software.

Creative Micro Designs (CMD)
P.O. Box 646
15 Benton Dr.
East Longmeadow, MA 01028
(800) 638-3263
(413) 525-0023 (questions/support)
Specialized hardware: hard drives, Jiffy DOS, RAMLink, RAMDrive, SID Stereo cartridge, and Swiftlink. CMD has also picked up the rights to *RUN* software and now distributes items once available through *RUN* and other software companies.

Creative Pixels LTD
P.O. Box 592
Library, PA 15129
64 games: *Escape Route—The Adventures of Eric Hawthorne, PI; Island of the Dragons;* and *The Wings of Circe.*

Dave's Computer
5259 Mayfield Rd.
Lyndhurst, OH 44124
(216) 646-1124
Refurbished hardware and a large selection of used software titles. No catalog available because of turnover rate. Call or write with specific requests.

Delphi
1030 Massachusetts Ave.
Cambridge, MA 02138
(800) 695-4005
Nationwide computer online service.

Disk O' Plenty
8362 Pines Blvd., Ste. 270
Pembroke Pines, FL 33024
(305) 963-7750
PD and shareware software. (Specify computer type.)

Diskoveries
P.O. Box 9153
Waukegan, IL 60079
For a copy of its catalog, send a stamp. OR send $2 for a catalog and disk.

Easy Books Distributing
P.O. Box 216
Iowa Park, TX 76367-0216
(817) 592-4727
Easy Books 1, a bookkeeping program for the 64.

8 Bit
P.O. Box 542
Lindenhurst, NY 11757-0542
(516) 957-1110
PD and shareware software. Also some commercial software and accessories.

Elliam Associates
P.O. Box 2664
Atascadero, CA 93423
(805) 466-8440
CP/M software—both commercial and PD, disk-copying service, and manuals. Send a self-addressed stamped envelope for a flyer or $1 for a catalog.

EMS Computer Services
4355 Kinney Rd.
Ludington, MI 49431
(616) 845-1540
New and used software and hardware. Free catalog. (Specify computer type.)

The FGM Connection
P.O. Box 2206
Roseburg, OR 97470
(503) 673-2234
Fun Graphics Machine, a graphics program for the 64.

FOG International
P.O. Box 1030
Dixon, CA 95620-1030
(916) 678-7353
The international CP/M user group. One-year membership is $40.

FRD Software
3487 E.Terrace
Fresno, CA 93703
GEOS graphics and utilities, scanned artwork, Doodle and Koala graphics. Send $1 for catalog.

Free Spirit Software
P.O. Box 158
Trafalgar, IN 46181
(800) 638-5757 (orders)
(812) 376-9964
Commercial software.

GEnie
P.O. Box 6405
Rockville, MD 20849-6403
(800) 638-9636
National computer online service. Has a good-sized 64/128 area.

GE Ridge Services
848 Rensselaer Ave.
Staten Island, NY 10309
Lotto Picker, a lottery program for the 64.

GEOWorks/Commodore Order Processing Center
5334 Sterling Center Dr.
Westlake Village, CA 91361
(800) 772-0001 ext. 1012
Super catalog! Full of useful information and answers to the most commonly asked GEOS questions along with the product listing.

The Grapevine Group
3 Chestnut St.
Suffern, NY 10901
(800) 292-7445
(914) 357-2424
Commodore chips, parts, power supplies, repair helps, and also some hardware.

Harvey & Associates
1409 Mill St.
Laramie, WY 82070
(307) 742-3275
Repairs. New and preowned software and hardware and accessories. For catalog, specify computer type and send $1 and a self-addressed stamped envelope. One dollar is refundable with first order.

Home Spun Software
P.O. Box 1064
Estero, FL 33928
PD and shareware software. International software. Ham radio utilities, high-quality arcade games. Some unique things found here! To receive a catalog, send a stamp.

Horse Feathers Graphics
N. 27310 Short Rd.
Deer Park, WA 99006-9712
(509) 276-6928
Western graphics and clip art.

House of Software
P.O. Box 282008
Columbus, OH 43228
(614) 871-1137
PD software.

Howard W. Sams & Co.
2647 Waterfront Parkway East Dr.
Indianapolis, IN 46214-2041
(317) 298-5400
Computerfacts (repair books). Send $5.95 for a complete catalog.

Jameco Electronics
1355 Shoreway Rd.
Belmont, CA 94002
(415) 592-8097
Electronic and computer parts.

Jason-Ranheim Company
3105 Gayle Lane
Auburn, CA 95603
(916) 878-0785
Specialized hardware such as an EPROM programmer, utility cartridges.

JP PBM Products By Mail
N. Sheridan Mall Postal Outlet
Box 60515
Downsview, ON
Canada M3L 1B0
Refurbished Commodore and new equipment. Catalog on disk for $2.

Jumpdisk
1493 Mountain View Ave.
Chico, CA 95926

Kaltec
Adjuntas, Puerto Rico 00601-0971
64 Scan-On-Printer system.

Kasara Microsystems
21-G Airport Rd.
Hilton Head Island, SC 29926
(800) 248-2983
(803) 681-5510
New and refurbished hardware, new software, and assorted products and accessories.

KC Enterprises
P.O. Box 627
Meriden, CT 06450-0627
(203) 634-0170
(203) 634-9359 (fax)
PD user group (software).

Keystone Software
P.O. Box 8369
Pittsburgh, PA 15218
Carries some commercial and PD/shareware software, Leroy's cheatsheets (keyboard overlays), printer ribbons, and a wide variety of continuous labels sizes.

KFPD
P.O. Box 470464
Los Angeles, CA 90047-0464
PD and shareware software for the 64/128/+4/16. Send a self-addressed stamped envelope for more information or $4 for a 64/128 catalog.

Living Proof
P.O. Box 80714
Minneapolis, MN 55408-8714
Specialized software: *I Paint* (an outstanding graphics program for the 128) and also *I Port.*

Loadstar (Softdisk Publishing)
P.O. Box 30008
Shreveport, LA 71130-0008
(800) 831-2694
(318) 221-8718
A monthly software magazine for the 64. A three-month trial subscription is $29.95. Also publishes *Loadstar 128,* a quarterly disk publication for the 128.

Lyben
P.O. Box 130
5545 Bridgewood
Sterling Heights, MI 48311-0130
(313) 268-8100
Computer supplies and accessories.

Mad Man Software
1400 E. College Dr.
Cheyenne, WY 82007-3298
(800) 34-MADMAN (orders)
(307) 632-1178 (info)
Software: *Messiah III, Master Archiver, The Compression Kit,* and others.

Master Software
6 Hillery Ct.
Randallstown, MD 21133
(410) 922-2962
Software such as *Tax Master* and *Help Master.* Specialized cables including an 80-column on composite monitor cable. A reset switch for the 64.

Maurice Randall
P.O. Box 606
Charlotte, MI 48813
Software: *Invoicewriter 2, GeoShell,* and *Dashboard 128* (GEOS).

MEI/Micro Center
1100 Steelwood Rd.
Columbus, OH 43212
(800) 634-3478
Computer supplies and accessories. Excellent printer ribbon selection (including the 1526).

Micro Storm Software
Box 1086
Sidney, MT 59270
Software: *Risers & Sliders* (game) and *S.E.C. Check Register 128* version 3.0

Mid Continent Computers
2332 N. MacArthur
Oklahoma City, OK 73127
(405) 946-2888
Used software.

Midwest Micro-Peripherals
6910 U.S. Rt. 36
E. Fletcher, OH 45326
(800) 552-8080
Computer supplies and accessories.

New Horizons Software
2253 N. Kansas Ave.
Springfield, MO 65803
Software: GEOS applications, Landmark Series, *Dualtop 64* and *128*.

North American Computer
147 W. Broadway
Waukesha, WI 53186
(414) 547-6222
Authorized Commodore repair shop.

Novastar Game Co.
P.O. Box 1813
Rocklin, CA 95677
(800) 352-2225
(916) 624-7113
Game software—mostly strategy, simulations, and D & D games.

Original Programs
P.O. Box 14801
Reading, PA 19612
Speciality software programs, many unique. Free catalog.

Parsec, Inc.
P.O. Box 111
Salem, MA 01970-0111
PD software, some commercial software: *Masterpiece Designs, Bellterm, 128 Handy Grabber, Handy Scanner, Page Fox.* Also publishes *Twin Cities 64/128.*

Performance Peripherals
5 Upper Loudon Rd.
Loudonville, NY 12211
(518) 436-0485
Specialized hardware: battery backups for REUs.

Poseidon Electronics
103 Waverly Pl.
New York, NY 10011
(212) 777-9515
Excellent CP/M library. Send $10 for complete catalog.

Powerdisk
6818 Lotus Way
West Jordan, UT 84084-7510
PD games, commercial software, disk drives, joysticks, and so on.

Public Domain User Group
P.O. Box 1442
Orange Park, FL 32073
(904) 264-3512
PD and CP/M software.

Q Enterprises
P.O. Box 77123
Washington, DC 20013-7123
Record Filer I, a database for home or office.

Quantum Link (Q-Link)/American Online Services
8260 Westwood Center Dr.
Vienna, VA 22180
(800) 827-8444
National computer online service for 64 owners.

Quincy Software
9479 E. Whitmore Ave.
Hughson, CA 95326-9745
GEOS software: *TOPDESK* and *DweezilDisks*

Random Magazine
7161 N. Maine
Clovis, CA 93611-8200
A bimonthly publication for the 64. Sample issue is $5.00. One-year subscription is $23.70. A companion disk is available.

RIO Computers
3310 Berwyck St.
Las Vegas, NV 89121
(800) 782-9110
(702) 454-0335
Assorted specialized hardware and software: *Advanced Art Studio, Handyscanner 64, MIDI 64, Advanced Music System, Deep Scan Burst Nibbler,* video digitizer, and genlock.

RM Software
P.O. Box 825
Maywood, NJ 07607
(201) 843-3116
(201) 843-8250 (BBS)
CNET 128 BBS.

RTH Compuware
805 Timber Ln.
Glenwood, IA 51534
(712) 527-3863
PD user group.

Schnedler Systems
P.O. Box 5964
Asheville, NC 28813
(704) 274-4646
Specialized items: digital I/O board, *MAE64* (assembler), *Symbol Master* (disassembler), and so on.

64 Disk Connection
4291 Holland Rd., Ste. 562
Virginia Beach, VA 23452
PD software. Send a stamp for a catalog OR $2 for catalog and disk.

Skyles Electric Works
P.O. Box 1984
Cupertino, CA 95015-1984
(800) 227-9998
(408) 737-1632
Lots of specialized hardware and software: IEEE Interfaces, cartridge expansions, 1571 fix ROM, parallel print cable, and so on.

The Soft Group
P.O. Box 111
Montgomery, IL 60538
(708) 851-6667
Video Byte (video digitizer), *Super Video version 5*.

Software Clearance Center
2025 Southwest Frwy.
Houston, TX 77098
(800) 825-1990
Commercial software closeouts.

Software Etc.
801 S. Greenville Ave.
Allen, TX 75202
(214) 727-9695
A consignment store that handles Commodore items.

Software Hut
313 Henderson Dr.
Sharon Hill, PA 19079
(800) 932-6442 (orders)
(215) 586-5701 (info)
(215) 586-5706 (fax)
Authorized repairs. Refurbished and some new hardware. New chips, parts, and software.

Software Support International
2700 N.E. Andersen Rd., Ste. A-10
Vancouver, WA 98661
(800) 356-1179
(206) 695-1393
Hardware, accessories, new and used software. Free catalog. If you want only the most recent listing of used software, send a self-addressed stamped envelope. Specify computer type.

SOGWAP Software
115 Bellmont Rd.
Decantur, IN 46733
(219) 724-3900
Specialized software: *Big Blue Reader 4.1* and *Bible Search 3.2.*

Spark's Electronics
5316 S. Ninth St.
St. Joseph, MO 64504-1802
(816) 238-5235
Financial software, *S.E.C. Check Register 128 version 3.0,* and more.

Storm Systems
464 Beale St.
W. Quincy, MA 02169-1307
GEOS software.

Sunrise Software
439 Park Ln.
Massapequa, NY 11762
Denny's Quest arcade/adventure game ($24.95) for the 64/128.

Tenex Computer Express
P.O. Box 6578
South Bend, IN 46660
(800) 776-6781
(219) 259-7051
New software, hardware, and accessories. Call for catalogue.

Terrapin Software
400 Riverside St.
Portland, ME 04103-1068
(800) 972-8200
LOGO and educational software based on LOGO.

Titus Software
20432 Corisco St.
Chatsworth, CA 91311
(818) 709-3692
New software: games and European imports.

TPUG
5334 Yonge St., Box 116
Willowdale, Ontario
Canada M2N 6M4
(416) 253-9637
User group supporting ALL Commodore computers. Bimonthly newsletter and access to an extensive PD software library.

Tycom
503 East St.
Pittsfield, MA 01201
Authorized Commodore repair center.

West River Audio Visual
1212 Oregon
Radio City, SD 57701-2006
(800) 272-1591
(605) 348-3389
Service and sales (parts) for Commodore computers. Refurbished hardware.

William Bacheler
3250 Trinity St.
Eureka, CA 95501
PD software for 64/128 and Amiga.

Zephorus Software
P.O. Box 463
Raymond, Maine 04071
(207) 655-4377
PD and shareware software for the 64/128.

Handi Hams System Care of Courage Center
3915 Golden Valley Rd.
Golden Valley, MN 55422
(612) 520-0515
Assists persons with disabilities in pursuing a hobby in radio.

The National Cristina Foundation
(800) CRIS-TINA
Matches computer donations with worthy causes.

SpeedCalc

Kevin Martin

Note: This article and program appeared originally in the January 1986 COMPUTE *magazine. It has never been printed in* Gazette. *(Some typographical errors were introduced when the article was ported to disk format; these errors have been corrected wherever I have found them. Graphical elements taken from the original article, included here to add some visual interest. –MS)*

Have you ever planned a budget for your home or office? If so, you probably used some sort of worksheet divided into rows and columns. Perhaps you wrote the months of the year along the top of the sheet and expenses along one side. After entering data for each category and month of the year, you could calculate total income figures by adding and subtracting numbers in each of the sheet's "cells."

That's a classic example of a worksheet. It lets you enter and organize data, then perform calculations that produce new information. A spreadsheet program is an electronic version of the familiar paper worksheet. Since it does all the calculations for you at lightning speed, an electronic spreadsheet is far more convenient than its paper counterpart. And spreadsheet programs offer built-in editing features that let you manipulate large amounts of data with a minimum of effort.

SpeedCalc is an all machine language spreadsheet program for the 64. Though relatively compact in size, it's fast and easy to use, and has many of the features found in commercial spreadsheet programs. Even better, the *SpeedScript Integrator* program (also included on this disk) lets you merge your *SpeedCalc* files into word processing documents created with *SpeedScript, COMPUTE's* popular word processor. Working together, *SpeedCalc* and *SpeedScript* make a powerful team. You can merge a chart of sales figures into a company report, create a table of scientific data for a term paper, and manipulate numeric information in a many other ways. In a sense, a spreadsheet program brings to arithmetic all the flexibility and power that a word processor brings to writing.

The *SpeedCalc* Screen

SpeedCalc uses the top of the screen as the command line. This is where *SpeedCalc* displays messages and asks you questions.

Screen lines 2-4 are the input buffer area. This is the work area where you enter and edit data. As you'll see in a moment, the input buffer also displays the data contained in the current cell.

The lower 21 screen lines are your window into the spreadsheet. Though your spreadsheet contains many rows and columns, only a few can fit on the screen at a time. By scrolling the screen back and forth with the cursor, you can move the display window to any part of the spreadsheet.

The *SpeedCalc* worksheet contains 50 vertical columns labeled with letters (AA, AB, AC, ... BX) and 200 horizontal rows numbered 1-200. The intersection of a row and a column is called a cell. Cells are where you store data. With 50 columns and 200 rows, the SpeedCalc spreadsheet has a maximum of 10,000 (50 x 200) cells. Due to memory limitations, however, only about a third of these can actually contain data. But you may spread out the data among all 10,000 cells if necessary, depending on the format you need.

If you don't like the spreadsheet's screen colors, they're easily changed with the special function keys. Press the f1 key to cycle through the 16 border colors until you find one you like. The f3 key changes the background color and the f5 key changes the character color.

Moving the Cursor

Each cell is identified with the letters of its column and the number of its row, For example, the cell at the extreme upper-left corner of the sheet is called AA1, since it's in column AA and row 1. The cell below that is AA2. Moving one cell to the right from AA2 puts you in AB2, and so on. (For the sake of clarity, this article uses uppercase letters for cell names. Note, however, that you must use lowercase letters such as aa1 when entering cell names from within *SpeedCalc*.

Your current position in the spreadsheet is shown by the highlighted cursor. The simplest way to move around the sheet is with the cursor keys, which work just as they do in BASIC. Another way to move the cursor is with the Home key (press Clr/Home without pressing Shift). Press Home once to "home" the cursor on the current screen; the cursor moves to the upper left cell. Press Home twice in succession to move the cursor to cell AA1, the home position for the entire sheet.

SpeedCalc also has a GOTO command for moving over a long distance. Press Ctrl-G (hold down the Control key and press G). The command line turns light blue and displays GOTO: followed by an underline cursor. The underline cursor generally indicates that SpeedCalc is waiting for data—in this case it expects the name of the cell where you wish to go. If you enter ba188 at this point, *SpeedCalc* moves the cursor to cell BA188, adjusting the screen window as needed. Take a few moments to practice moving around the spreadsheet with all three methods—you'll be using them a lot. In a later section, we'll discuss how to change the size and format of a cell.

Keyboard Commands

SpeedCalc offers many different commands, a few which are entered by one key. However, most commands are entered by pressing Ctrl along with some other key. Ctrl-G, as you've seen, is the GOTO command. Ctrl-A displays the amount of free memory available, and so on. The most drastic command is Ctrl-X, which exits *SpeedCalc* and returns you to BASIC.

Since leaving the program effectively erases all data in memory, *SpeedCalc* asks ARE YOU SURE Y/N? before shutting down. To cancel the command and return to the spreadsheet, type N and press Return.

A few commands require you to press three keys at once. This sounds more awkward than it is in actual practice, since two of the three keys are Shift and Ctrl. For example, the command to shift between automatic and manual recalculation is performed by pressing Shift-Ctrl-R. (hold down Shift and Ctrl, then press R). The accompanying table at the end of this text lists all the *SpeedCalc* commands. We'll be discussing each command in more detail later.

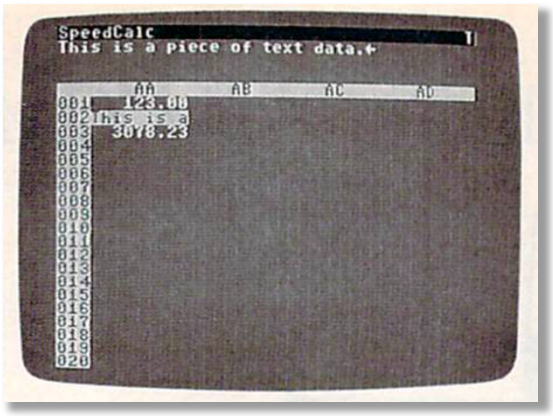
Three Data Types

Before entering any data, you must know what kind of data *SpeedCalc* accepts. There are three different types: numbers, text, and formulas. Let's look at each type in turn.

1. ***Numeric Data*** consists of numbers—the basic stuff that spreadsheets work with. *SpeedCalc* has a few simple rules for numeric data. A number must be a decimal value (base 10, not hexadecimal) composed of one or more digits from 0-9, with an optional plus or minus sign. A decimal point is also optional. If you include any other characters in numeric input, *SpeedCalc* treats the entire input as text data (as explained below). Thus the numbers 123, .001, and -65535 are valid numeric data. The numbers 65,535 (which includes a comma) and 312 Main Street are treated as text labels.

For example, let's enter the number 123 in cell AA1. No special commands are required to enter data: just move the cursor to AA1 and begin typing. While you're entering the number, it appears only in the input buffer near the top of the screen. As soon as you press Return, the number appears in AA1 and the letter N appears in the upper right of the screen. The N signifies numeric, meaning that *SpeedCalc* has accepted the entry as valid numeric data. Move the cursor to a vacant cell, then move it back to AA1. The input buffer displays whatever data is found in the cell under the cursor. When the current cell is empty, the buffer is empty as well.

As you can see, pressing Return enters a data item into the current cell. You can also end the input by pressing the cursor key. The data is entered as if you had pressed Return, and the cursor moves in the indicated direction. This feature is handy for entering a lot of data: Simply type the entry, move the cursor to the next cell, enter more data, and so on.

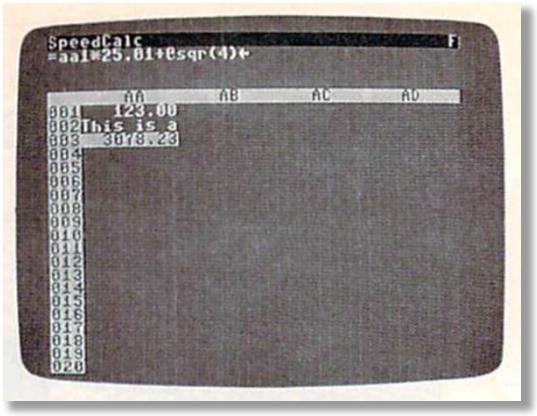


2. ***Text data*** is not "data" in the strict sense, since *SpeedCalc* doesn't use it in calculations as it does numbers and formulas. Text data is there only to help humans understand what the other data means. Text may consist of comments, titles, column headings, subheadings, or whatever you need to interpret the numbers and formulas. As an example, move the cursor to cell AA2 (the cell under AA1) and type the following line. (The quotation marks are optional.)

```
"This is a piece of text data."
```

Note that both uppercase and lowercase letters are acceptable. You can use the Del key to erase mistakes while you're typing. When you press Return, *SpeedCalc* displays T for text in the upper-right corner. In this example, the cell isn't long enough to accept the entire text, so only the leftmost portion appears in cell AA2. But even though you can't see all the text, it's there. Move the cursor to another cell, then move it back to AA2. As soon as you return to AA2, *SpeedCalc* displays all the text in the input buffer.

3. ***Formula data*** is a mathematical expression or formula. It may be as simple as 2 + 2 or as complex as your imagination (and mathematical prowess) allows. The first character in a formula must always be an equal sign (=). If you omit this symbol, *SpeedCalc* either signals an error or treats the data as text. The true power of a spreadsheet is that the formula in one cell can refer to data in another cell. This is easier to demonstrate than to explain. Move the cursor to cell AA3 and type the following line.




```
=aa1*25.01+@sqr(4)
```

As soon as you press Return, *SpeedCalc* displays F (for formula) in its upper-right corner and puts the result of the formula (not the formula itself) in AA3. If AA1 contains 123, the value 3078.23 appears in AA3. In plain English, this formula means "multiply the contents of cell AA1 by 25.01 and add the square root of 4." Before we examine the formula more closely, here's a quick demonstration of what makes a spreadsheet such as powerful tool. Move the cursor back to AA1 and press Ctrl-R. The command line displays the message RECALCULATION IS ON, meaning that *SpeedCalc* now automatically recalculates the entire sheet whenever you make a change. Now change the number in AA1 to 456 (simply move to that cell and start typing). The new result (11406.56) automatically appears in cell AA3. We'll explain more about automatic recalculation later.

Mathematical Operators

These symbols can be used as operators in a formula:

Operator	Function
+	addition
-	subtraction
*	multiplication
/	division
^ (up arrow)	exponentiation
=	equality

One factor that affects formulas is precedence, or the order in which mathematical operators are performed. In *SpeedCalc,* formula operators have the same precedence as BASIC—the same as in general math.

The first operators to be evaluated—those with the highest precedence—are those enclosed in parentheses. Where one set of parentheses encloses another, the expression in the

innermost set is evaluated first. The next operators to be evaluated are exponents. Multiplication and division have equal precedence; both operators are lower than exponentiation. Addition and subtraction have the lowest precedence of all. The mnemonic "My Dear Aunt Sally"—(MDAS: Multiplication, Division, Addition, Subtraction) is a reminder of mathematical precedence.

To take one example, *SpeedCalc* evaluates the formula =5*(8+3*-2)^2-10/+2 as the value 15, just as in ordinary math. Note how the result is affected by the plus and minus signs before the 2s.

Functions

Formulas may also include any of the functions listed here:

@abs()	absolute value
@atn()	arctangent
@ave()	average of a block of cells form: @ave(xxn:xxn)]
@cos()	cosine of argument in radians
@exp()	complement of log, gives e^x (e=2.7182318...)
@int()	integer (numbers to lowest whole number)
@log()	natural logarithm base e (log of 0 or a negative number is illegal)
@sgn()	sign (-1 for negative numbers, 0 for 0, and +1 for positive)
@sin()	sine of an argument in radians
@sqr()	square root (root of a negative number is illegal)
@sum()	sum of a block of cells [form: @sum(xxn:xxn)]
@tan()	tangent of an argument in radians (@tan(.5*pi) is illegal
pi	value of pi (3.14159265)

All the functions except pi begin with the @ symbol are followed by parentheses. (To use pi by itself in a cell, precede it with an equal sign, such as =pi.)

Within the parentheses of a function you may use a number or a formula. For example, the formula =@sqr(4) generates the square root of 4. The formula =@sqr(aa1) return the square root of whatever value is in AA1. Note that the argument (the value within the parentheses) of the functions @tan(), @sin(), and @cos() must be expressed in radians; the result of the function @arc() is expressed in radians.

The function @int() generates an integer (whole number) by rounding to the next lowest number. For positive numbers, this is the same as dropping the fraction, but for a negative number like -4.3 the next lowest number is actually 5.

The function @ave() calculates the mean average of the values in a block (group) of cells. The function @sum() calculates the sum of a block. Both functions require that you define the block so that *SpeedCalc* knows which cells to include in the calculation. This is done by putting the two cell names separated by a colon in parentheses. The first cell name defines the upper-left corner of the block, and the second defines the bottom-right corner. For instance, @ave(aa1:ad20) calculates the average of all cells from AA1 to AD20. The function @sum(aa1:ad20) calculates the sum of all cells from AA1 to AD20, and so on. An error results if any cell in the block is blank or contains text data.

Math Notes

SpeedCalc uses the same ROM routines for math as BASIC. Therefore, it follows almost the same rules and has the same limitations. Numbers are accurate internally only to nine digits—although you can enter long numbers and view them exactly as you entered them, only the first nine digits are used in the calculations. If you enter a very long number (more than 36 digits), your input is ignored, and the cell reverts to its former state. You can also enter long numbers in the form 1.23E+05 (scientific notation). Note, however, that *SpeedCalc* itself never uses scientific notation. It converts all numbers to their full length, so long numbers actually use more memory than shorter numbers.

Beware of math errors such as division by zero, square root of a negative number, tangent of pi/2, logarithm of zero or a negative number, and overflow conditions like 2^300. *SpeedCalc* detects these errors and displays *ERROR* in the cell of an illegal calculation.

Editing the Sheet

Editing is a very important spreadsheet function. The simplest way to change what a cell contains is to move to the cell and start typing. The old data in the cell is replaced with whatever you enter. For instance, to replace the data in AA1 with the number 456, move to that cell, type 456, and press Return or exit with a cursor key.

Press Ctrl-B (think of blank) to erase what's in the current cell. You can also clear a cell by typing as space and pressing Return, but this uses some memory. A truly blank cell does not use any memory.

To erase everything in the sheet, press Shift-Clr/Home. before carrying out this drastic operation, *SpeedCalc* ask you to confirm by pressing Y or N.

In some cases, only a minor change is needed. Edit mode lets you change data in a cell without retyping the entire entry. To activate edit mode, move to the desired cell and press Ctrl-E. In this mode the up/down cursor is disabled, and the left/right cursor key moves within the input buffer. Erase unwanted characters with the Del key. Typing in edit mode inserts new characters in the line. Everything to the right of the new character moves to the right one space (unless the buffer is already full). Since the cursor keys have a different function in edit mode, you cannot use them to end the input. Press Return to enter the new data and escape from edit mode.

As you may have learned already, *SpeedCalc* displays *ERROR* in a cell when you enter an erroneous formula. The usual cause is that you have made a typing error in that cell, or the formula refers to text or an empty cell. A line of asterisks (**********) signals that a number is too large to be printed in the cell. Though these messages appear in the cell area, no data is lost. You may move to the affected cell, view its contents in the input buffer, and make whatever correction is needed

Recalculation

The recalculation feature is the very core of *SpeedCalc.* As you know, entering a piece of data causes *SpeedCalc* to perform a calculation and put the result in the cell under the cursor. In most cases, the new data relates to data in other cells, so you'll eventually want to recalculate

the entire spreadsheet as well. This can be done in two different ways: manually and automatically.

To recalculate the spreadsheet manually, press the back arrow key, at the upper-left of the keyboard. *SpeedCalc* begins at AA1 and recalculates the entire every cell that contains data,placing fresh results wherever needed. *SpeedCalc* displays the message RECALCULATING while its busy.

If you switch to automatic mode, *SpeedCalc* automatically recalculates the entire spreadsheet each time you enter new data or edit what exists. When you press Ctrl-R, *SpeedCalc* changes the recalculation status and displays it at the top of the screen. If automatic recalculation was turned off before, it is now on (and vice versa). If you aren't sure which mode you're in, press Shift-Ctrl-R; *SpeedCalc* displays the recalculation mode without changing it.

Automatic recalculation can be fun to watch in a large spreadsheet. Every time you make a change, you can watch the new results ripple all the way down the screen. However, the more data your spreadsheet contains, the longer it takes to update the entire sheet. For this reason, you may wish to turn off automatic recalculation most of the time. Recalculate with the back arrow key whenever you need to view the results.

One problem with recalculation arises from the order in which cells are calculated. Because only one cell can be calculated at a time, you must sometimes recalculate the entire spreadsheet two or three times to get correct results in every cell (this is common to all spreadsheet programs). For instance, say you have a formula in AA1 that refers to a formula in AB15. When *SpeedCalc* calculates AA1, it must use the existing data from AB15—which is probably out of date, since the formula in AB15 hasn't been recalculated yet. To avoid this problem, you should press the back arrow key several times before printing a spreadsheet or saving it to disk.

SpeedCalc offers a number of other features. Before experimenting with them, you should spend some time typing in a hypothetical spreadsheet—perhaps a fictitious yearly budget—to become thoroughly familiar with the basic commands covered so far. Most importantly, create formulas, using all the operators in different combinations. Try doing things that you know will cause errors. Then correct the errors in edit mode, and so on. It takes a thorough grasp of the fundamentals to get the most out of *SpeedCalc's* advanced features.

Change Type and Format

The default (normal) format for numeric data is flush right with rounding to two decimal places. In other words, the number is displayed in the rightmost part of the cell, with two numbers after the decimal point. text and formulas are flush left (shown in the leftmost part of the cell). *SpeedCalc* offers several commands for changing cell formats.

Change Format (Ctrl-F). This command changes the location of the data in the cells and the number of decimal places. When you press Ctrl-F, SpeedCalc displays the question FORMAT: Left, Center or Right justify? in the command line. Press L, C, or R to move the data to the left, center, or right of the cell.

Change Number of Decimal Places (Ctrl-D). This command lets you specify the number of digits displayed after the decimal point. The default value is 2, convenient for dollar amounts, but you may change it to anything from 0-15. If you choose zero decimal

places, any number in that cell is rounded off to the nearest integer (whole number). A setting of 15 is special: The number in that cell is not rounded off at all. Instead, *SpeedCalc* displays the number exactly as entered or calculated from a formula. Watch out for one feature of Ctrl-D: It also resets the cell to right justification, so you may need to change this with Ctrl-F.

Width (Ctrl-W). The width command changes the width of the entire column of cells. Move the cursor to any cell within the desired column and press Ctrl-W. When *SpeedCalc* displays the prompt Width, you should respond with a number from 4-36. The entire screen is redrawn to accommodate the new format, and may look very different, depending on what value you choose. For instance, if you increase a column's width, the rightmost column of the former display may disappear. *SpeedCalc* only displays as many complete columns as it can fit onto a screen. If you decrease the width of a column, you may see asterisks where numbers used to be (indicating that the cell is now too narrow to display the entire number). To get rid of the asterisks, expand the column as necessary.

Global Format (Shift-Ctrl-F). This is the same as the ordinary format command, but operates globally, changing every cell in the spreadsheet instead of just one. To alert you to the difference, *SpeedCalc* changes the color of the command line to blue.

Global Width (Shift-Ctrl-W). This is the global version of the width command. The command line turns light green to signal the difference. Every column in the sheet changes to the designated width.

Global Decimal (Shift-Ctrl-D). This command lets you change the number of decimals displayed in the entire sheet. The default for the sheet is two decimal places. Every cell changes to the designated setting, and the new setting becomes the default for future entrees.

Macroediting

After typing a large spreadsheet, you may decide to make a major change. You may want to add new data somewhere in the middle, delete a section, or move a group of cells from one location to another. *SpeedCalc's* macroediting (large scale editing) commands simplify such operations, affecting entire blocks of cells at once. A block is simply a group of cells connected in rectangular fashion: You can define it as a single cell, a row or column, or any rectangular area within the sheet.

There are two ways in which macro commands can work: verbatim or relative. To make a simple example, say that cell AA2 contains the formula =aa1*5 and you want to move its contents to cell AB2. When you do this in verbatim mode, AB2 contains an exact copy of what was in AA2 (=aa1*5). Note that the cell name used in the formula does not change: The formula still refers to AA1. If you perform the same operation in relative mode, the cell name in the formula is adjusted to fit the new location. In this case, AB2 would contain the formula =ab1*5.

Copy (Ctrl-C). The copy command copies a block of cells into a new location with disturbing the original ells. Place the cursor in the upper-left corner of the block that you want to copy, then press Ctrl-C. SpeedCalc changes the command line to purple and prompts you to move the cursor to the lower-right corner of the block that you want to copy. Once the cursor is in place, press Return.

Now *SpeedCalc* prompts you to move the cursor to the position where you want to put the block: This is the upper-left corner of the new position. Once the cursor is there, press Return again. The new data replaces whatever was contained in the designated cells.

Note that if you define an impossible block (for instance, moving the cursor to the upper-left of the original position, rather than below and to the right), S*peedCalc* does not copy any data. You can use this trick to escape from Copy if you press Ctrl-C accidentally. Another escape is to press Return twice while the cursor remains on the original cell.

Move (Ctrl-M). This command works like copy, but it fills the original cells with blanks. Though *SpeedCalc* has no Insert command, you can use this command to make space for new data in the middle of a spreadsheet. Simply move everything below the insertion point down as far as you need. To cancel, press Return twice while the cursor is on the same cell.

Relative Copy (Shift-Ctrl-C). This form of the copy command adjusts to the cell names used in formulas within copied blocks (see explanation above.)

Relative Move (Shift-Ctrl-M). This is the relative form of the move command. Cell names in formulas are adjusted to reflect the move.

Memory Management

SpeedCalc leaves 10,752 bytes of memory (10.5K) available for data. As noted earlier, *SpeedCalc* spread your data around over a much large range of cells than you can actually fill with data. The extra space is provided to give you full control over the final format. For example, you could have a 15 x 150 spreadsheet and leave some elbow room for move and copy operations.

Because memory is limited, you should keep track of how much is free while using the program. Press Ctrl-A to display the amount of free memory. we suggest limiting your spreadsheet to 1296 cells (36 rows by 36 columns). If you have filled nearly all of free memory, you may have to break the spreadsheet into two smaller sheets.

Although *SpeedCalc* checks the amount of available memory, and displays an error message if you run out of memory, you should be careful not to exhaust free memory. Any move or copy operation in process will be aborted if sufficient memory is not available.

Disk Operations

SpeedCalc has four disk commands that allow you to save a spreadsheet to disk, load it, display the disk directory, and send commands to the disk drive. The directory command is the simplest to use. Press Ctrl-4 (think of the dollar sign, as in LOAD"$",8 to list the directory from BASIC). The screen clears and the directory is displayed. Press Return to return to the normal screen. You may pause the directory display with the space bar.

To save a spreadsheet to disk, press the f8 function key. *SpeedCalc* prints SAVE: on the commands line followed by an underline cursor. Enter a valid Commodore filename and press Return. (If you change your mind and decide not to save the file, press Return with entering a filename.)

To load a saved file again from disk, press the f7 key. Again, *SpeedCalc* asks for a filename. *SpeedCalc* files are saved a PRG (program) files, but do not load as normal program files. *SpeedCalc* uses special header bytes to identify a *SpeedCalc* file. If you try to load anything other than a valid *SpeedCalc* file, you'll see a special message NOT A SPEEDCALC FILE.

You can send Commodore disk commands to the drive with Ctrl-^—press Ctrl and the ^ (up arrow) key together. *SpeedCalc* prompts you to enter a disk command. You need not enclose the command in quotation marks or enter a ,8 extension. For example, enter Ctrl-^ and then I0 to initialize a disk. Consult your disk drive manual for information about disk commands.

Printing

SpeedCalc lets you print data to three different devices: to the screen for previewing output, to a printer for permanent documentation, or to a disk file for integrating the data with another program.

The preview your spreadsheet on the screen, press Shift-Ctrl-P, then press S (screen output) when prompted. Naturally, the display will look odd if your sheet is wider that 40 columns. Try to think of each pair of 40-column lines as one 80-column printed line.

To print a hardcopy of your spreadsheet, Press Ctrl-P. If your printer is configured like most, this should produce a satisfactory printout. This command sends output to the printer as device number four with a secondary address of seven (uppercase/lowercase on most systems).

NOTE: Before using this command, you must position the cursor below and to the right of the block of cells that you wish to print.

The upper-left corner of the printout starts at cell AA1. The entire width you define by this position is used. Therefore, do not try to print overly wide spreadsheets that won't fit on the paper. If you want to print a spreadsheet wider than 80 columns, many printers have a condensed mode that lets you fit 132 characters on a line. You can change this by switching internal DIP switches, or by sending CHR$ code from BASIC before running *SpeedCalc.* Many printers will switch to condensed mode if you send the following command.

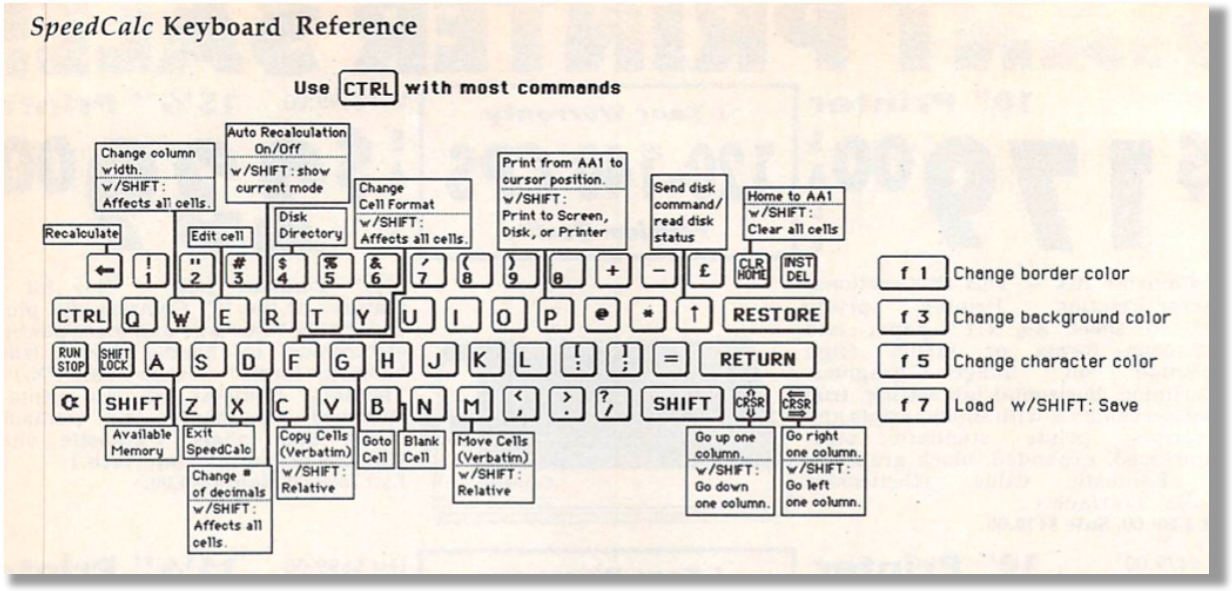
```
OPEN 4,4: PRINT#4,CHR$(15): CLOSE 4
```

To send output to a printer with a device number other than four or a secondary address other than seven, enter Shift-Ctrl-P, then enter the device number and the secondary address when prompted. During a printout, you can pause the output by pressing Shift or Shift Lock. The screen border turns white and printing ceases until you release Shift. Press Run/Stop to abort printing.

You can also print *SpeedCalc* data to disk for use with terminal programs, databases, or word processors, including *COMPUTE's SpeedScript.* Select the D option after pressing Shift-Ctrl-P, then enter the filename that you wish the new file to have. The disk file is an exact Commodore ASCII (PETSCII) image of what would go to the printer.

Note that printing to disk creates a different file than saving to disk. You should save files that you wish to reload into *SpeedCalc,* and print files that you wish to convert for *SpeedCalc* or other programs. while you may pause this operation by pressing Shift as with the printer output, do

not Run/Stop to abort printing to a disk. This may create a Splat (unclosed) file which can be safely removed only by validating the disk.



***SpeedCalc* Commands**

Ctrl-A	Available memory
Ctrl-B	Blanks (erases) a block cells
Ctrl-C	Copy block verbatim
Ctrl-D	Set number of decimals
Ctrl-E	Edit current cell
Ctrl-F	Format a cell
Ctrl-G	GOTO selected cell
Ctrl-M	Move block verbatim
Ctrl-P	Print sheet
Ctrl-R	Recalculate toggle
Ctrl-W	Change column width
Ctrl-X	Exit *SpeedCalc*
Ctrl-4	Disk directory
Ctrl-^	Send disk command
Shift-Ctrl-C	Copy block relative
Shift-Ctrl-D	Change decimal mode of all cells
Shift-Ctrl-M	Move block relative
Shift-Ctrl-P	Print to screen, disk, printer
Shift-Ctrl-R	Display current recalculation mode
Shift-Ctrl-W	Change width of all columns
Shift-Clr/Home	Clears spreadsheet
f1	Change border color
f3	Change background color
f5	Change character color
f7	Save *SpeedCalc* file
f8	Load *SpeedCalc* file
(back arrow)	Recalculate spreadsheet

Programs

SpeedScript Integrator
Kevin Martin

Convert SpeedCalc *files for use with* SpeedScript.

SpeedCalc sends data to the printer in simple plain-vanilla form. That may be fine for personal use, but if you're creating a document for others to view, you may want special features such as boldface, underlining, and so on.

Since *SpeedScript, COMPUTE's* popular word processor, already offers an easy way to access these features, no attempt has been made to include them in *SpeedCalc.* All that's needed is a simple program to convert *SpeedCalc* files into a form that *SpeedScript* can load. Then you can edit the file with *SpeedScript* as you would any other document—inserting printer control codes, reformatting the text, merging it with other text, and so on.

Here are the steps to convert a *SpeedCalc* file for use with *SpeedScript.*

1. After creating a spreadsheet file with *SpeedCalc,* print it to disk by pressing Shift-Ctrl-P, then D for disk. (Remember, this is different from saving a file to disk.)
2. Exit *SpeedCalc,* then load and run *Integrator.* The program prompts you to enter the name of the *SpeedCalc* file that you printed to disk. Then it asks you to enter the name of the *SpeedScript* file that you want to create (of course, this filename should be different from the first). The *Integrator* constructs a *SpeedScript*-loadable file from the *SpeedCalc* file.
3. After the *Integrator* is finished, load and run *SpeedScript,* then load the new *SpeedScript* file as you would any *SpeedScript* document. The data appears onscreen, ready to be edited in any way you wish.

If you already have *SpeedScript's Sequential File Converter,* you can use this program's Commodore ASCII-to-*SpeedScript* option to convert *SpeedCalc* files. This option works like the *Integrator.*

Batch Programming
Steven Reid

Note: This program requires SpeedScript *to operate.*

Most of us have a few programs that do simple but very useful things without interfering with our BASIC programming. Unfortunately, these programs are usually written in machine language and are difficult to create. Another drawback is that they must reside in memory, and if two such programs use the same area of memory, they will not work together. Of course, I wouldn't be writing about this without offering a solution: batch files.

Batch files, common with the IBM crowd, are ordinary text files. They can be created with a word processor, yet they contain program instructions. Such programs can run completely from disk, so they don't take up any memory and won't disturb your BASIC programs unless they are designed to.

The batch interpreter I am presenting here is fairly simple; it sends much of the program to BASIC. It's as though you typed in each instruction in immediate mode. You may wonder why bother with batch files? Why not just type in the command? Some commands can get quite complex. For instance, have you ever tried to read the disk error channel from immediate mode? Even though you can't use INPUT# or GET#, it can still be done with the following lines.

```
OPEN15,8,15:POKE781,15
SYS65478:FORI=0TO-1STEP0:SYS65487:SYS65490:I=PEEK(780)=13:NEXT:SYS65484
CLOSE15
```

Do you really want to commit something like that to memory? Of course, you could just tack the usual error reading onto the program in memory and call it when you need it, but batch files let you do almost anything from immediate mode. And once you create a batch file, you won't have to retype it.

Required Programs

To use batch files, you will need two of the programs presented, Readbat.49152 and Batch Util. They are on the flip side of this disk. There are some other related files there as well. The program Bufbat.Maker is optional. The other programs, *Convert, Partram, Time, Textlist,* and *Disk,* are sample batch files. I'll discuss them later.

Readbat.49152 is the batch interpreter, the program that runs your batch files. This is the main program, and it's written in machine language.

The other program you will need is Batch Util. You won't need it very often, but you will need it. Batch Util is written in BASIC.

Also, you will need *SpeedScript,* Gazette's word processor, to enter any batch programs and convert them to the sequential files that *Readbat* understands.

Create a Batch File

To start, load and run Batch Util. Press 4 to select the Make Empty File option. You will be asked for a filename. Insert a work disk on which to write your batch files, preferably a disk containing *SpeedScript.* Enter EMPTY FILE for the filename. The program writes a short file to disk. This file isn't really empty; it contains several *SpeedScript* control codes.

Reset your computer and then load and run *SpeedScript.* Insert the disk containing EMPTY FILE and load it with f7 as you would any *SpeedScript* document. On return, when the document appears you will see several control codes at the top of the screen. These codes must be at the beginning of all of your batch files for them to work properly. Cursor past the end of the control codes and enter the following as a *SpeedScript* document. Press Return at the end of each line.

```
'set screen colors
poke53280,6
poke53281,0
printchr$(159)
]e
```

At the end of the file, press Ctrl- and then R. Then type 0. This sets the right margin at the end of the file to 0, so you don't waste space in the file you are about to create. Insert a disk and press Ctrl-Shift-P and then D to print the *SpeedScript* file to disk as a sequential file. For the filename, enter COLORS. *SpeedScript* will create a SEQ file on that disk, a file that Readbat can run.

Reset your computer and then load Readbat.49152. Enter NEW to reset pointers and then type SYS 49152 to activate the program. Once *Readbat* is activated, it deletes the activation portion of itself to make room for storage. Any further calls to 49152 will just return you to BASIC.

Readbat is patched through the Kernal CHRIN and BASIC CHRGET routines. It's a wedge that you can call from immediate mode. Insert the work disk that contains COLORS and enter the following command.

```
@COLORS
```

The computer will read the file COLORS from disk. If the file isn't on that disk, you will get a FILE NOT FOUND error message. If all is well, you will see the POKE and PRINT commands printed to the screen and the screen colors changing accordingly.

You've just created your first batch file. Now, let's go back to the original *SpeedScript* file and examine the commands.

```
'set screen colors
```

This line starts with an apostrophe, so the entire line up to and including the Return is ignored. The apostrophe is like REM. This is the best way to put remarks into a batch file.

```
poke53280,6
```

This poke is sent to BASIC as though you had just typed it in. It sets the border color to dark blue.

```
poke53281,0
```

This command sets the screen color to black.

```
printchr$(159)
```

This sets the text color to cyan.

```
]e
```

The right square bracket tells *Readbat* not to send the line to BASIC; *Readbat* is to interpret that line itself. The E is like the BASIC command END. It means that the program should

return to BASIC and begin accepting input from the user once again. There are several other commands accessed through the] character; they will be explained later.

This is an extremely simple batch file. Later, I'll show you how to create programs as complex and useful as the *Convert* and *Disk* samples.

The Examples

I have included several sample programs to demonstrate the usefulness of batch files. They were created the same way by starting with the Empty File commands and *SpeedScript*. Here's what they do.

Convert – This program is a hex/decimal number converter, probably the most useful example given. To convert a number, enter the following.

```
@CONVERT,number
```

Number is the number you wish to convert. If it has the $ prefix, it will be accepted as a hexadecimal number and will be converted to decimal. Without a $, it will be taken as decimal and converted to hex.

Partram – This program divides BASIC RAM into two separate partitions; each can hold a different program and variables. Partition 1 occupies $0801 to $7000. Partition 2 starts at $7001 and goes to the top of memory, wherever that was when the program was initialized. This program must be initialized before you try to change partitions. To initialize it, enter the following.

```
@PARTRAM,INIT
```

Afterwards, you can switch between partitions easily with these commands.

```
@PARTRAM,1
```
or
```
@PARTRAM,2
```

On your first call to partition 2, you should perform a NEW to clear pointers; otherwise, the computer could lock up. To restore the full amount of BASIC RAM, enter this command.

```
@PARTRAM,OFF
```

Partram runs slowly unless you're using *Bufbat*.

Time – This program sets or displays the time in the TOD clock (CIA chip #2). Before you can use it, you have to set the time.

```
@TIME,hhmmssf
```

In setting the time, hh is the hour, mm is minutes, ss is seconds, and f is the am/pm flag, either A or P. Hours, minutes, and seconds should be two digits each, and the entire string should total seven characters. After you set the time, you can display it with the following command.

```
@TIME
```

Textlist – This short program will become more useful when you begin writing your own batch files. It will list any Commodore ASCII file to the screen; it can even list itself. Here is the format to use it.

```
@TEXTLIST,filename
```

FILENAME is the name of the text file that you wish to list.

Disk – This program allows you to send commands to the disk drive. To run the program, enter the following command.

```
@DISK
```

When you see the prompt, enter the command you wish to send. If you enter a dollar sign with an optional search pattern, the program will list the disk directory without interfering with the BASIC program in memory. You can pause the listing with the space bar or end it with the Stop key. To quit the program and return to BASIC, enter Q. This program can be quite slow and is practically useless unless you are using *Bufbat.*

The *Bufbat* Program

Executing a batch file directly from disk can be slow, so I wrote *Bufbat.* It's a program that patches itself into *Readbat* and reads a file into memory before running it. Load and run the program Bufbat.Maker. It will write a program called BUFBAT.828 to your work disk. This is the program you will actually use. Load BUFBAT.828, enter NEW, and then type the following.

```
SYS828,address
```

Address is the start of the buffer. SYS828,40960 works very well in most cases, placing the file under BASIC ROM. Now, when you call your batch files, they will be read into memory before running. *Bufbat* almost always makes a file run faster. To deactivate the program, enter SYS 831. *Bufbat* is not really necessary, but the increased speed is well worth it.

So far, this article has barely scratched the surface of what batch programming can do. In part 2, I will explain *Readbat's* numerous other functions.

More on Programming

You will probably find these sample programs quite useful, but with a little effort, you can learn to write your own. This next section goes into more detail on how to use *Readbat* commands.

You may be confused by the cryptic codes that are mixed in with the more familiar BASIC commands. After reading this, you should be able to understand the *Readbat* commands, modify the samples, and write your own utilities. Take your time, there's a lot here to cover.

Passing Parameters

Try the following batch file. Remember, enter it as a *SpeedScript* document with all of the control codes mentioned and print it to disk with the filename PARAM.

```
‘parameter passing
rem a [0
rem b [1
rem c [2
rem d [3
rem e [4
]e
```

Remember that]e marks the end of the file. I also explained the apostrophe is used to write batch file remarks. I still use the BASIC REM statement, though. Unlike the apostrophe, *Readbat* will print BASIC commands such as REM to the screen. This lets you see exactly what *Readbat* is sending to BASIC. Reset your computer and load *Readbat;* enter SYS49152 to activate. Insert the disk with PARAM and enter the following command.

```
@PARAM,ME,MYSELF,AND I
```

Beneath the Ready prompt, you will see the five remarks from the batch file, but they won't be exactly like the ones you entered. The five square brackets and their associated numbers are gone and, except for the last one, have been replaced by parts of the line that activated the batch file. This is how the activation line is structured.

```
@parameter0,parameter1,parameter2...
```

This sets continues to parameter 9. Parameter 0 is always the filename PARAM. In our example, parameter 1 is ME, parameter 2 is MYSELF, and parameter 3 is AND I. The last REM shows an empty parameter 4. Parameters 4 - 9 weren't included in the activation line, so they are null. You may now understand parts of the examples, such as those found in *Convert.*

```
za$=mid$("[1",2):zb=0
```

and

```
za=[1/4096:zb$=""
```

In the first line, parameter 1 is used in a BASIC string, so just about anything will be accepted. With the second line, parameter 1 is expected to be a number, such as 119 or 1024, but as you can see, 2*3 would also be accepted within the expression. If, however, you were to send something that would make that line unacceptable to BASIC, an error would result.

For instance, if parameter 1 were something like POKE, BASIC would produce a syntax error. Because of the error, that line would not be executed. *Readbat* would be unaware of the error, and the batch file would continue to chug along, possibly producing other errors because that line had been ignored. As a result, the end output would be meaningless.

Readbat Variables

Readbat has 26 simple variables, A through Z. These are separate from those in BASIC. They can be used within a batch file much like parameters and can hold an integer between 0 and 255. If variable C is set to 125, a line such as

```
I=TR+[C
```

would be sent to BASIC as the following.

```
I=TR+125
```

Variables can't be set from the activation line the way parameters can; they have to be set within the batch file. To set variable C equal to 125 you have to use hexadecimal in the following format.

```
][C=7D
```

Just about every *Readbat* command uses hexadecimal. If you can't convert 125 to $7D in your head, don't worry. The *Convert* sample will do the job for you.

The variables are stored in memory from $C003-$C01C (49155-49180) and occupy one byte each. Here is another way to set C equal to 125.

```
poke49157,125
```

This pokes the variable directly. Unfortunately, this makes programs rather difficult to read. This method does have one advantage in that BASIC can perform complex math functions. *Readbat* only offers some simple addition and subtraction, such as the following examples show.

`][a=12`	sets A equal to $12 (18)
`][a+40`	adds $40 (64) to the current value of A (A=A+$40)
`][q-0a`	subtracts $0A (10) from the current value of Q
`][c=[d`	sets variable C equal to variable D
`][a+[b`	adds variable B to the current value of A

You may want to convert a given parameter (a number, of course) to a variable. To set variable A equal to the number given in parameter 1, enter the following.

```
poke49155,[1
```

Variables are important within a batch file, especially with *Readbat's*]t command. More on that shortly.

Readbat Commands

All of *Readbat's* commands begin with a right square bracket. Look at the variable setting/addition/subtraction. It starts with a right bracket, suggesting a command, and then the left square bracket, something like the BASIC LET statement. You've seen the]e command that marks the end of the program. You can also use it to put a string into the keyboard buffer. Here's how it's done.

```
]e "@time" 0d
```

That would put six characters into the keyboard buffer, the five in quotes plus the 0d, which is the same as CHR$(13), a Return. So, this batch file would end by calling the Time batch file.

Advanced Information

There is a routine within the *Readbat* interpreter that I call GETBYTE ($C108, 49416 decimal). This routine is used in almost every *Readbat* command. If you understand how it works, you may gain a better understanding of what is and is not acceptable.

Each time GETBYTE is called, it returns with one byte pulled from the batch file. The 0d shown in the]e line above would be sent back as one byte, $0D. Spaces are ignored, so 0 d is the same as 0d.

If GETBYTE finds a variable, the variable would be peeked and that value would be sent back. If GETBYTE encounters a quotation mark, it goes into a quote mode. In this quote mode, 0d would be taken as two bytes, 0 and d. The spaces would also be recognized.

Variables are also sent differently. Rather than being sent as one byte, they are sent as one to three ASCII digits. In quote mode, parameters can be included and are sent as a string of bytes. The only way to turn off quote mode is to include a closing quote. Always close your quotes! If you don't, quote mode may be on when it's supposed to be off and vice versa. You can still leave the closing quotes off your BASIC commands such as OPEN and LEFT$, because GETBYTE is never called with those commands. Just be careful with Readbat commands that begin with a right square bracket.

Commands

Here's a list of every *Readbat* command with a detailed explanation for each.

`]g` — GOTO a new point in the program. Example: `]g labelname1`

There are no line numbers in batch files, so you have to mark specific points within the program by placing labels there. Labels work like the remarks beginning with an apostrophe, except labels are recognized by the]g command and begin with an @ sign.

For instance, @labelname1 would be ignored during execution but would be recognized by]g. The]g command works by closing and reopening the batch file and then doing a byte-by-byte search through the file, looking for the label. Because of the sequential nature of text files,]g is slowest when jumping to the end of a file.

With *Bufbat* activated, the search still works beginning to end but is far faster because the search is through memory rather than the disk. If the entire file has been searched and the label can't be found, the program will break with an UNDEFINED STATEMENT error. One warning—there are no end-of-file markers in RAM, so *Bufbat* will search forever if the label doesn't exist. Of course, that won't be a problem if you don't]g to a nonexistent destination.

`]t` – TEST is like the BASIC IF/THEN. With]t, a colon (:) serves as the THEN marker. If the statement is true, execution resumes just past the colon. If the statement is false, execution resumes just past the return marking the end of the line. Here's the syntax.

```
]t string = string :then true
```

If either string is greater than 20 characters, an error will occur. The first string is padded with zeros to 20 characters, and only the number of bytes in the second string are compared. If you are testing greater-than (>) or less-than (<), only the first byte of each string is compared. There is no testing for not-equal (<>). Each string can be any series of bytes recognized by Readbat's GETBYTE routine. Here are some examples.

```
]t 41 42 43 = "abc" :true
]t "abcde" = "abc" :true
]t "abcde" = "abc" 00 :false
]t "abc" = "abc" 00 :true
]t "[1" = "end" :acceptable
]t "[1" = "end" 00 :acceptable
]t [a = 0d :acceptable
]t "[a" = "13" 00 :same as above
]t 2e < 2f :true
]t 2e > 2e :false
]t 2e 22 > 2e 21 :false (2e is not greater than 2e)
```

The]t command can be used to test the result of a BASIC comparison. Look at this example.

```
poke49155,abs(statement)
```

This works if the statement is any valid BASIC comparison, such as in$<>"abc" or sqr(i)>j*(2+j). If the result of the comparison is true, the A variable (49155) will contain a 1. If the comparison is false, A will be 0. Here's a better example.

```
poke49155,abs(in$<>"y")
]t [a=01 :in$="n"
```

That example amounts to IF IN$<>"Y" THEN IN$="N". Replace IN$="N" with a *Readbat* command, such as]g, or even another]t, and it can be very useful.

`]e` – END tells *Readbat* to close the batch file and return to BASIC.]e also allows you to set the contents of the keyboard buffer, and automatically sets the value to $C6 (198).

Be careful. If the string is more than ten characters, the Kernal file table will be overwritten. If no string is given, the keyboard buffer and its counter are left alone. Here's an example.

```
]e "@part2" 0d
```

This would cause the current batch file to end and would run the file part2. This is a very effective method of chaining batch files or just executing one final BASIC command. Chaining can be quite powerful because several small files will often run faster than one large one if the file is run from disk. Remember, a program run from disk can be hundreds of kilobytes in size.

]d – DISPLAY will enable or disable printing. It must be followed by a 0 or a 1. A 0 disables printing. Almost every batch file includes a]d0 at the beginning, so BASIC commands and the READY prompt don't appear on the screen.

Unfortunately, the BASIC PRINT and PRINT# commands are also disabled. If you need to send something to the screen or other device, you should use]o (explained later) or temporarily re-enable printing with]d1. You can make the printing of the commands invisible by setting the text color equal to the background color.

```
]p a 0286
]p b d021
]s 0286 [b
]d1
print#15,in$:rem  do printing1
]d0
]s 0286 [a
```

This example sets the cursor color to the background color so the PRINT# command and the READY prompts are invisible. It then restores the cursor color. The *Disk* example uses such a method when working with the command channel.

]o – OUTPUT works like the BASIC PRINT and PRINT# commands, except]o will work after]d0.]o must be followed by a single-digit number and then the string to send. If the number given is 0, the string will be printed to the screen. If the number is other than 0, it will be taken as a file number, and the string will be sent to that file. Because]o uses a one-digit number, it can only use files 1-9. You probably won't want to open anything higher. Here are some examples.

Printing a message to the screen.

```
]o0 "hello, joe!" 0d
```

Sending data to disk.

```
open3,8,2,"textfile,s,w"
]o3 "one,two,three,four,five"
```

After any]o command, [z contains the status variable. Don't use the usual BASIC status variable. It will be changed as the batch file is read.

]i – INPUT will get one byte from the keyboard or an open file. It must be followed by a letter A-Z representing the *Readbat* variable in which the data will be stored. A single-digit number should follow the variable. If the number is 0, *Readbat* will wait for a keypress and then place the ASCII value into the given variable. If the number given is not 0, it will be used as a file number, and the byte will be taken from that file. After reading from a file, the [z variable will contain the value of ST. The *Disk* sample uses this command when reading a disk directory.

]p – PEEK will place the contents of a memory location into a variable. Here's an example.

```
]p a d021
```

This places the background color ($D021) into the [a variable.

`]s` – STORE is like BASIC's POKE command, except it uses hexadecimal and can store a series of bytes.

```
]s 0500 01 02 03 "987"
]s d900 01 01 01 01 01 01
```

That will display "abc987" on the screen in white and will be visible even if your 64, like mine, fills color memory with the background color.

`]c` – CALL works like the BASIC SYS command. Like SYS,]c uses $030C-$030F (780-783) to hold the 6502 A, X, Y, and status registers. With]c, you can set these values with the command, instead of using a separate store instruction. Here are some examples.

`]c 033c` is the equivalent of SYS828.

`]c bdcd 01 ff1` will print 511 to the screen, unless you have used]d0.

`]c fff0 00 [y [x 00` will move the cursor to the position specified by the X and Y variables.

Batch File Errors

Eventually, a bugged batch program will halt with the message BATCH FILE ERROR AT BYTE followed by a hexadecimal number and then a BASIC error message. Write down the hexadecimal number and then load and run *Batch Util.* Press 1 for Find Error, and you will be prompted for a filename. Enter the name of the file you were running. You will then be prompted for a location. Enter the hexadecimal number you wrote down. The file will be listed to the screen.

At some point, the file will begin listing in reverse video (rvs on). Press space to pause the listing and look at the line where it changed. Once you know where the error is, go back to the original *SpeedScript* file and correct it and then print the corrected version to disk.

Other Uses

I originally wrote *Readbat* to load all of the programs I like to have in memory. You'll probably want to try it; just place all of the LOAD and SYS commands into a batch file. I used to use BASIC and machine language programs for this, but I would rather modify text than a machine language source file. The BASIC CLR, RUN, and NEW commands, as well as most fastloaders, will purge any open disk files, so use *Bufbat* to get around this. The only program I've had problems activating is *Bassem.* Parts of the program have to be disabled temporarily. Here's how it's done.

```
load"bassem",8,1
new
]s 87f0 ea ea ea ea ea ea
sys34782
]s 87f0 8d 26 03 8e 27 03
```

You can run batch files from a different drive; just poke the device number into 49154 ($C002).

If you have a program that stores strings in memory, it's very easy to change the string. For instance, I have a function key redefiner, and if I want to change the definition of the f7 key, I just call a batch file that contains the line]s 7cd2 "[1".

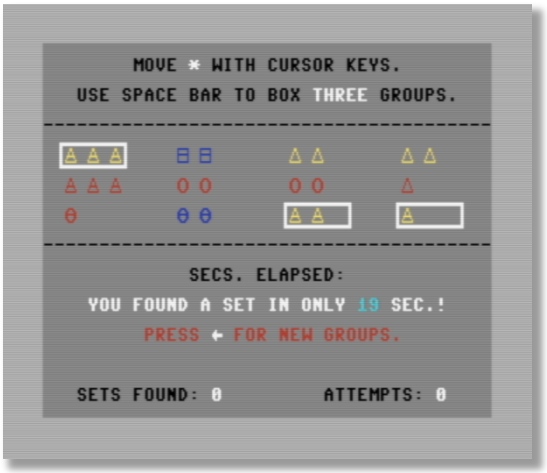
I also have a batch file that I use to save BASIC programs: just use @save,prog. Then the file PROG is erased, and the BASIC program in memory is saved with the filename PROG. There are many possible uses for batch files; you'll probably discover a few of your own.

I think I've said enough here for you to write your own batch programs. But don't expect to become a master right away; it will take time. After all, it's a completely new programming language.

Steven Reid is 16 years old and has been programming since he was 10. He lives in Squamish, British Columbia, Canada.

Setz!

Larry Cotton



The documentation for this mind teaser is contained within the program itself.

Instructions

Your objective: to find a *set* from 12 groups of figures as fast as possible. (My objective: to *time* your efforts!)

A set always contains *three* groups. Each group can have 1, 2, or 3 figures. Each group's figures can have up to 4 characteristics: quantity, color, shape, and fill.

However, more than one characteristic can vary! (e.g. a set with varying color and quantity, or a set with varying shape, fill, and color, etc.)

In other words, a set's characteristics *must* be either *identical* or *completely different.* For example, a set can contain 3 red groups; or 1 group each of red, yellow, and blue, but it cannot contain 1 red group and two blue groups! Looking at it another way, for any 2 groups there is always only 1 group which could complete the set.

You will be picking groups by pressing *the space bar.* After you find a set, the *computer* will decide whether it's legal or not.

One last note! There will *always* be at least one set on-screen! But if you can not find it soon, press [back arrow] for new groups because *you are being timed!* You should be able, with practice to average about 30 seconds per set.

Trail Blazer
Gus Vakalis

Trail Blazer is a colorful fast paced machine language arcade/strategy game for the 64. The object of the game is to collect five gemstones that appear at random locations while avoiding the bouncing—but deadly—happy faces and your own trail.

Trail Blazer is written in machine language, it loads and runs like a BASIC program. You can pause and upause the game by pressing the Run/Stop key, and you can reset the game by tapping the Restore key.

You begin *Trail Blazer* with three lives. You earn a free life every 5000 points up to a maximum of nine lives at any one time.

Use a joystick plugged into port two to control your man. Once you begin moving your man will remain in motion, leaving a multicolored trail behind him with each step. The trail starts out at only 50 sections long but grows by ten sections with each new screen level. It can grow up to a maximum of 250 sections. You'll want to avoid getting tangled in the trail but that gets harder to do as you advance through the levels.

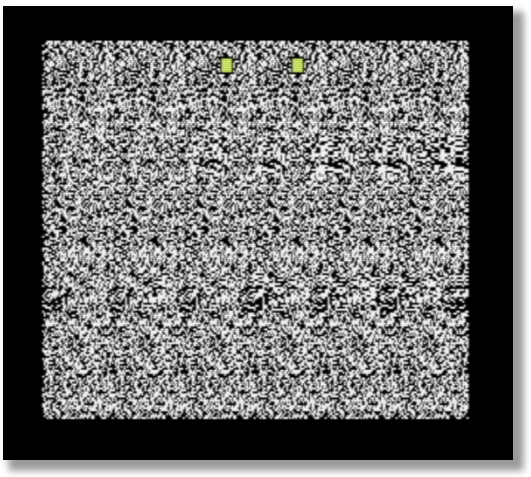
Gemstones appear one at a time at random locations about the screen. When you've picked up a gem, you'll receive 100 points and another gem will appear at another location. If you're lucky, you'll have a clear path to it. When you've collected all five gemstones, any time remaining on the timer will be added to your score and you'll advance to the next level with a longer trail.

The strategy in *Trail Blazer* is to lay your trail so that it doesn't hinder your progress in collecting the gemstones. That's pretty simple to do when your trail is only 50 sections long, but you'll need a sound strategy when you're dragging 250 sections of trail behind you.

In addition to avoiding your trail and the walls of the maze, you'll also have to watch out for those five happy faces. They bounce relentlessly around the screen, and any contact with the yellow dudes is fatal. Fortunately, you can control the speed of the happy faces. Skill level 1 is the easiest, with the happy faces bouncing at an orderly pace. Watch out for level 5 though, it's not for those who get dizzy watching a tennis match.

Stereogram 128

Don Klich



There are two basic ways to create the illusion of a three dimensional image on a two dimensional surface. The first requires the simultaneous taking of two pictures of the subject from slightly different positions, usually about six inches apart. This picture pair is then positioned so that an observer can see only one picture with one eye and the other picture with the other eye. The brain then merges the two images, creating what appears to be a three dimensional picture. The antique stereopticon and later stereo film viewers are examples of this technique.

Another approach to creating three dimensional images is the Random Dot Stereogram technique. That's the approach taken by Stereogram 128. You have a choice of several designs that you can turn into a 3-D graphic. When you make your selection from the program menu, the basic design is first drawn on the screen. This takes a few seconds.

Then a vertical band of randomly generated dots (pixels) is printed, scanning from left to right across the screen. These pixels print over the image that's on the screen. If, as the pixels are being duplicated, a portion of the image is encountered, a slight shift occurs in the pixel being printed.

When the band of random pixels encounters the image, your monitor will blank. Do not be alarmed, this is normal. The monitor will stay blank while the vertical band is printing over the image. This entire process takes several minutes, so be patient.

When the stereogram is complete, you'll see what appears to be a screen of random pixels with two yellow squares near the center. As you view the resulting stereogram, both eyes apparently see the same mosaic surface, but slight differences create a depth effect. The two yellow squares are spaced exactly the width of the random band. This is called the interocular distance.

Seeing is Believing

To view the screen, stare at the yellow squares and allow your eyes to "unfocus" until you perceive three equally spaced squares. It may take a while to get the hang of this. When you reach this state, hold it for about five seconds without refocusing your eyes, then look down at the rest of the screen or paper and in some cases at several different levels.

The variable S in the program which controls the depth effect can vary between +10 an -10, with negative values causing the image to appear below the plane and positive values above the plane.

Create Your Own Images

Stereogram 128 supplies a choice of seven image selections, each of which can be printed with any printer having graphic capabilities. To print a stereogram, press P and the Return key when a screen is completed.

Interested programmers can create their own 3-D pictures by writing two related subroutines for each screen. The first creates the image. Code for selection 1 is found on lines 360 through 420. When designing your image make sure that it is more than 48 pixels (six characters) from the left edge.

The second subroutine determines the depth value of S to be used for each image pixel. This can be a fixed value as in selection 2 (line 520) or vary dynamically as in selection 1 (lines 370-420). Each of the paired subroutines are called by lines 130 and 180 respectively.

Generally, the larger the image and/or the more complicated the calculation of S, the longer it will take to create the screen. Though it takes a while to create each screen, the results are worthwhile. This is especially true of selections 4, 5, 6, and 7.

Peagsis

Cleveland M. Blakemore

Pegasis is a public domain program that is discussed in "PD Picks." It does not contain its own separate documentation.

To learn more about this program before running it, be sure to read "PD Picks" found in the Columns menu on this disk.

Ghoul Dogs

Melvin Finstemocker

Ghoul Dogs is a public domain program that is discussed in "PD Picks." It does not contain its own separate documentation.

To learn more about this program before running it, be sure to read "PD Picks" found in the Columns menu on this disk.

An associate program on the flip side called *Ghoul Colours* lets you change the colors of the different elements in *Ghoul Dogs.* After you've made your color choices, enter SYS 49152 to restart *Ghoul Dogs* with these new colors.

Gazette Disk Bonus

SIGN UP A FRIEND AND GET A FREE GAZETTE DISK!

That's right. Sign up a friend to a one year's subscription to *Gazette Disk* and we'll extend your subscription by an extra issue.

Sign up two friends, get two free disks. If your friend signs up for two years, we'll see that you get two extra disks.

You probably know many Commodore owners who do not subscribe to Gazette Disk. Show this disk to them and see if they don't agree that 12 issues for only $29.95 is a great deal.

User group members, you can assign the extra disk to your group and any other subscriber of your choice. Impress your friends with your generosity!

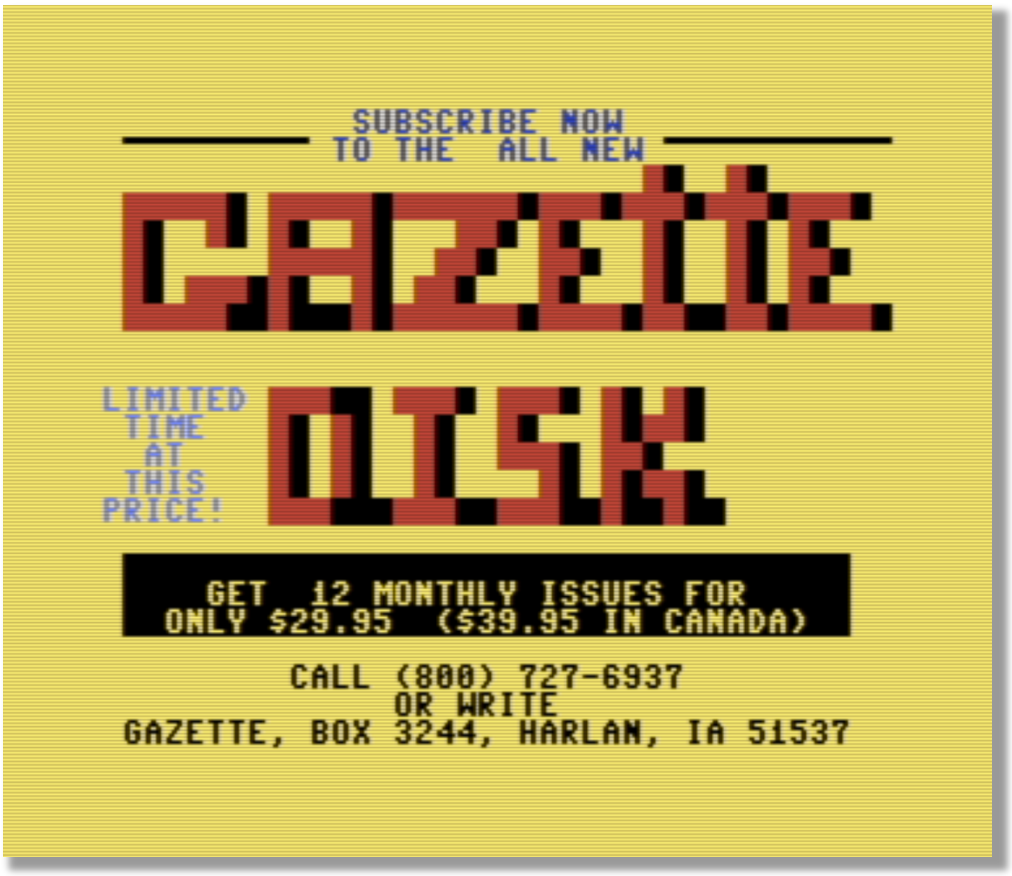
Call (800) 727-6937 and use your VISA or MasterCard, or just say, "Bill me!"

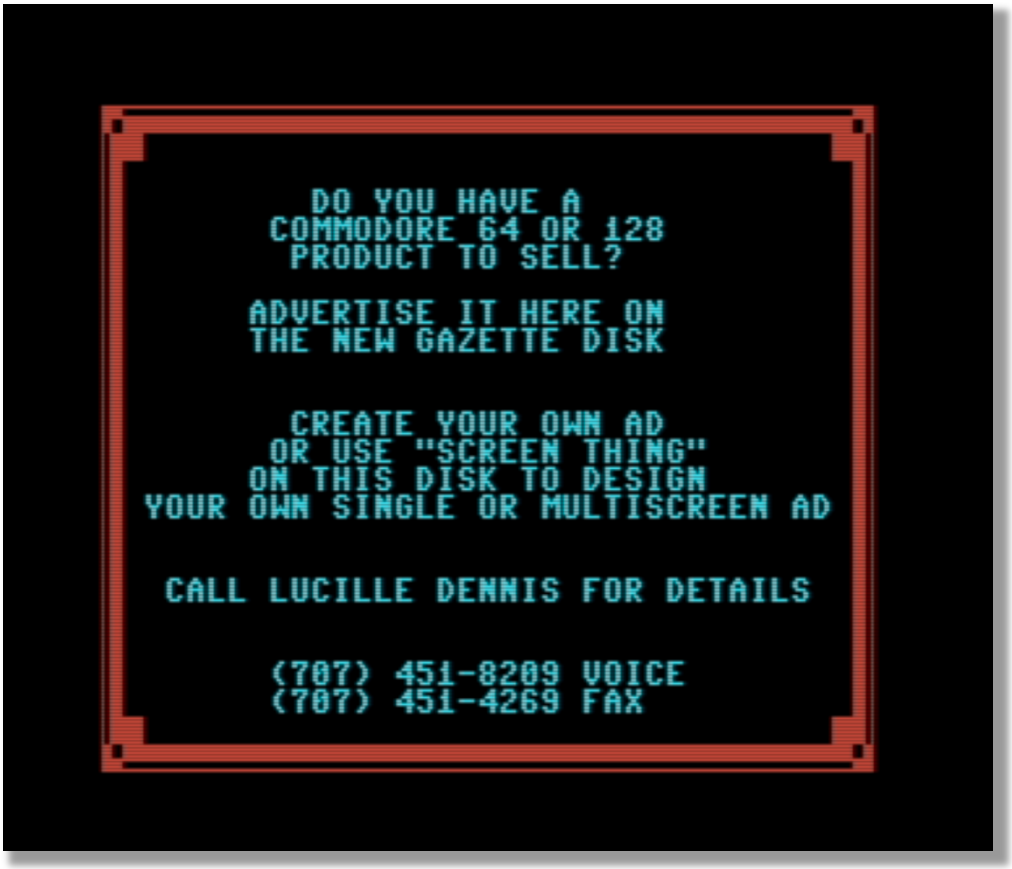
If you prefer, just print out the following page and send a check or money order for $29.95 ($42.75 in Canada, $39.95 in other countries) to

COMPUTE GAZETTE DISK
P.O. Box 3244
Harlan, IA 51593-3244

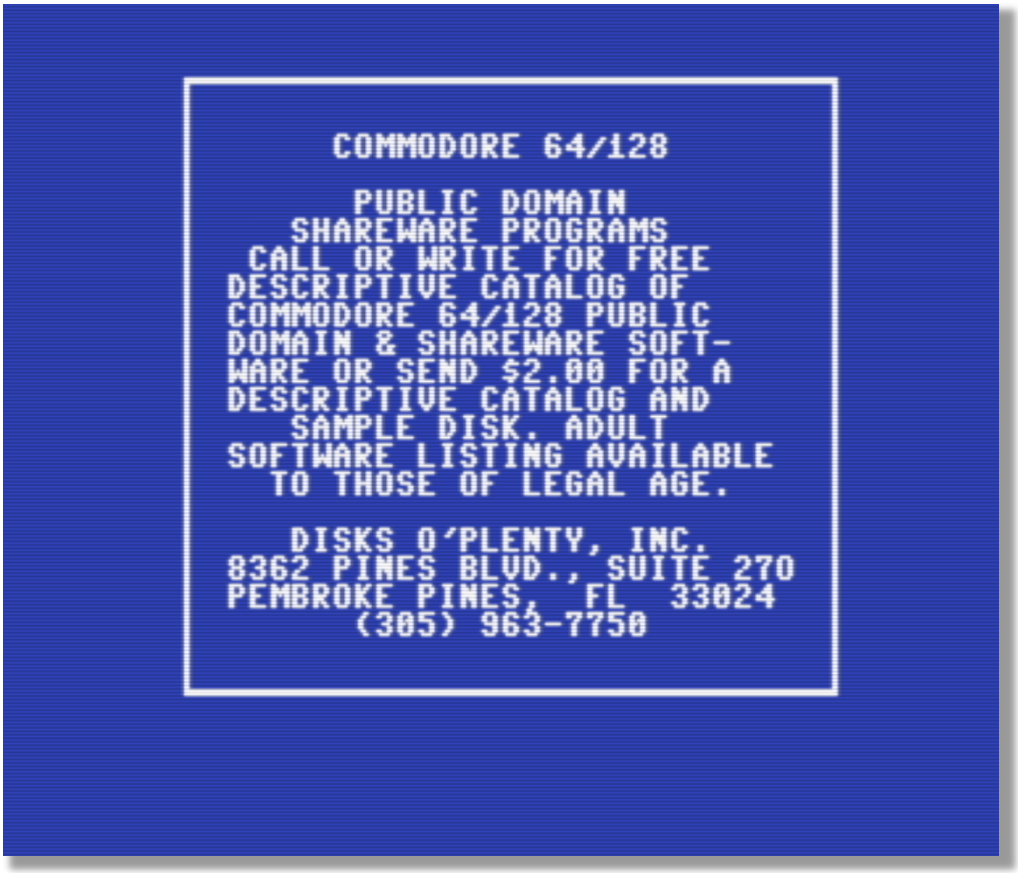
```
Key: HSSF3
New subscriber's
 Name:
 Address:
 City
 State/Province:
 Country

[ ]Payment enclosed       [ ]Bill me

Charge  [ ]VISA           [ ]MasterCard

Account #
Expiration date

Signature

Add the free disk to this subscription:  Name:
 Address:
 City:
 State/Province:
 Country:
```

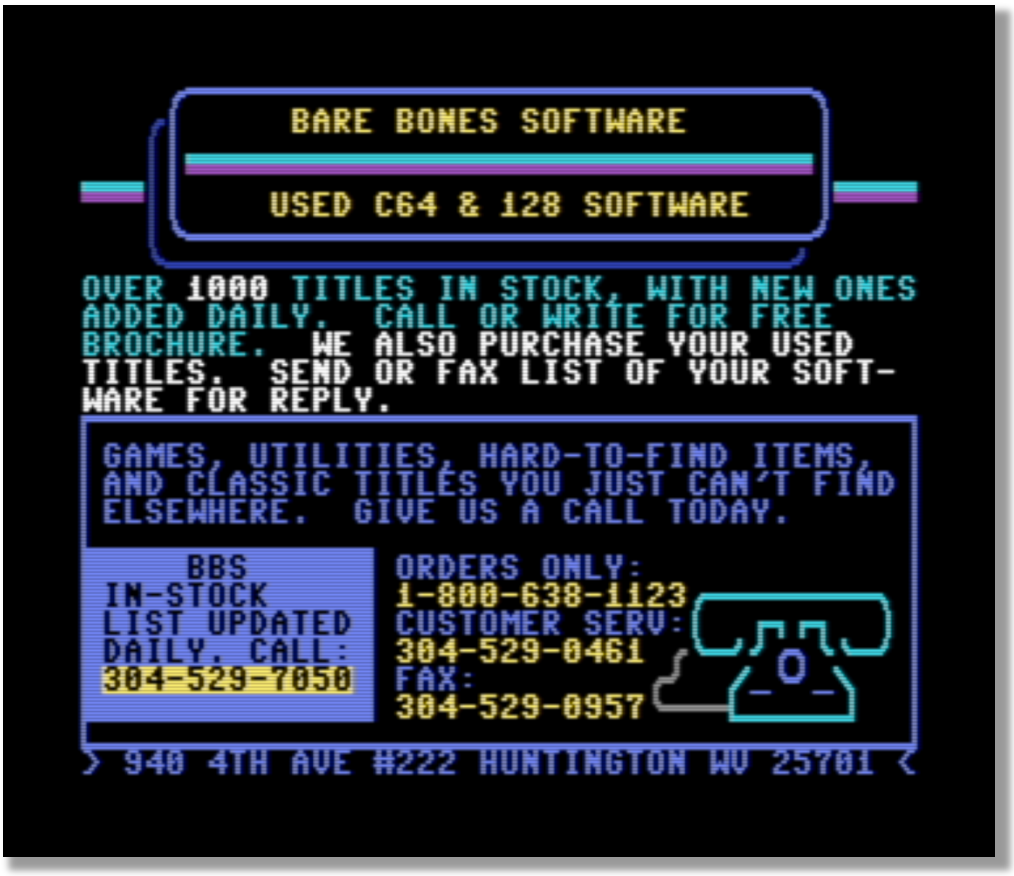
This price is valid through March 15, 1994.


SUBSCRIBE NOW
TO THE ALL NEW
GAZETTE
LIMITED
TIME
AT
THIS
PRICE!
DISK
GET 12 MONTHLY ISSUES FOR
ONLY $29.95 ($39.95 IN CANADA)
CALL (800) 727-6937
OR WRITE
GAZETTE, BOX 3244, HARLAN, IA 51537

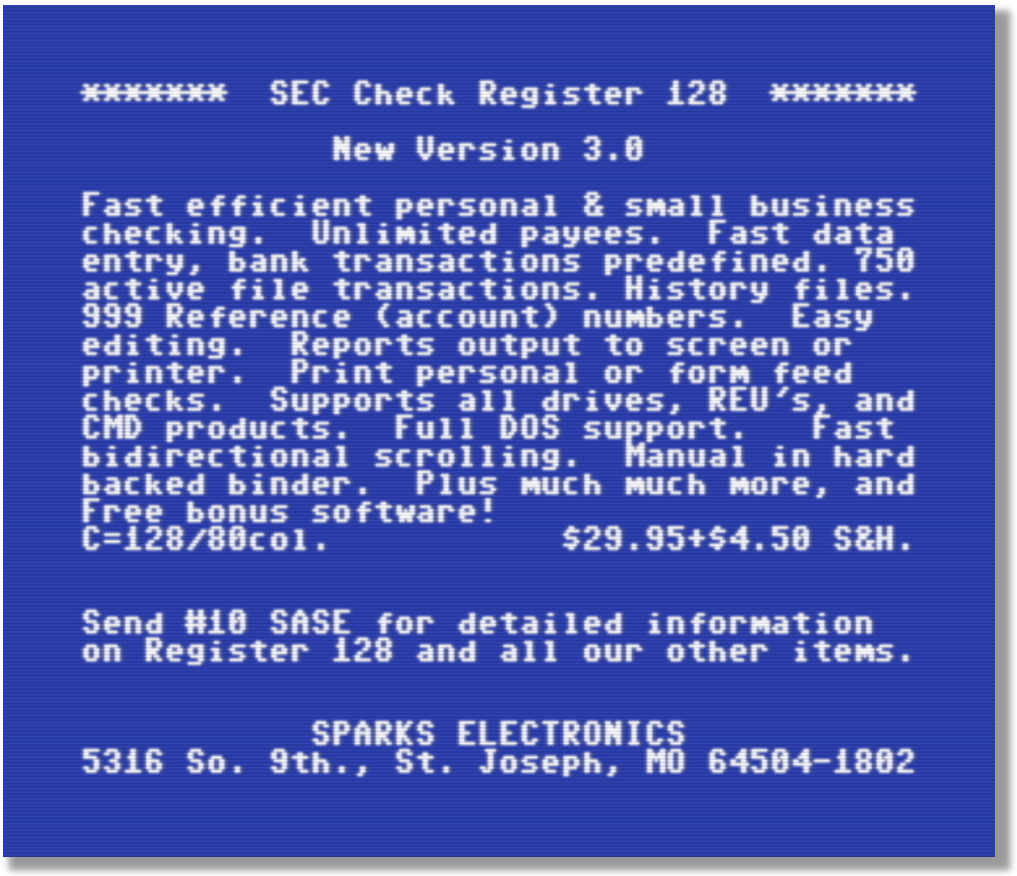

DO YOU HAVE A
COMMODORE 64 OR 128
PRODUCT TO SELL?
ADVERTISE IT HERE ON
THE NEW GAZETTE DISK
CREATE YOUR OWN AD
OR USE "SCREEN THING"
ON THIS DISK TO DESIGN
YOUR OWN SINGLE OR MULTISCREEN AD
CALL LUCILLE DENNIS FOR DETAILS
(707) 451-8209 VOICE
(707) 451-4269 FAX

REMEMBER ...
GAZETTE HAS CHANGED ITS FORMAT, BUT IT STILL WANTS TO PURCHASE YOUR BEST PROGRAMS AND ARTICLES.
SEND A DISK THAT CONTAINS YOUR PROGRAM AND DOCUMENTATION, PLUS A HARD COPY OF THE DOCS TO
GAZETTE SUBMISSION
324 W. WENDOVER AVE., STE. 200
GREENSBORO, NC 27408
NOW THAT WE'RE ON DISK, WE CAN USE YOUR LARGER PROGRAMS!


COMMODORE 64/128
PUBLIC DOMAIN
SHAREWARE PROGRAMS
CALL OR WRITE FOR FREE
DESCRIPTIVE CATALOG OF
COMMODORE 64/128 PUBLIC
DOMAIN & SHAREWARE SOFT-
WARE OR SEND $2.00 FOR A
DESCRIPTIVE CATALOG AND
SAMPLE DISK. ADULT
SOFTWARE LISTING AVAILABLE
TO THOSE OF LEGAL AGE.
DISKS O'PLENTY, INC.
8362 PINES BLVD., SUITE 270
PEMBROKE PINES, FL 33024
(305) 963-7750


BARE BONES SOFTWARE
USED C64 & 128 SOFTWARE
OVER 1000 TITLES IN STOCK, WITH NEW ONES ADDED DAILY. CALL OR WRITE FOR FREE BROCHURE. WE ALSO PURCHASE YOUR USED TITLES. SEND OR FAX LIST OF YOUR SOFTWARE FOR REPLY.
GAMES, UTILITIES, HARD-TO-FIND ITEMS, AND CLASSIC TITLES YOU JUST CAN'T FIND ELSEWHERE. GIVE US A CALL TODAY.
BBS
IN-STOCK
LIST UPDATED
DAILY. CALL:
304-529-7050
ORDERS ONLY:
1-800-638-1123
CUSTOMER SERV:
304-529-0461
FAX:
304-529-0957
940 4TH AVE #222 HUNTINGTON WV 25701

DENNY'S QUEST I
EXPLORE THE LAND MAP TO SEEK OUT THE EVIL GHOST IN THE CEMETERY. 6 SEPARATE AREAS INCLUDE ICY CAVES, A DESERT, THE CEMETERY, SECRET ITEMS/AREAS ETC... IN THIS COLORFUL NEW ARCADE ADVENTURE.
PRICE: $19.99 (INCLUDES SHIPPING)
FOR MORE INFORMATION OR TO ORDER DQ I (CHECK OR MONEY ORDER, U.S. CURRENCY ONLY PLEASE) WRITE TO:
SUNRISE SOFTWARE
PHILIP CILLO
439 PARK LANE
MASSAPEQUA PARK, NY 11762
OR CALL (516) 541-3902



Computer Bargain Store Ad

Huge selection of new and used Commodore hardware, software, games, and accessories. Hard to find items. Excellent prices. Example: Like-new (tested) Commodore 128D computers with built-in 1571 disk drive $179 + $10 shipping and handling (within the continental U.S.).

These are compatible with Commodore 64 & 128 hardware, software, and accessories and have a 30 day guarantee. These units came to us without original boxes, manuals, system disks, or TV/monitor hookups. Price includes CPU, power cord and keyboard. An incredible price for an incredible machine. Limited supply. Telephone orders taken between 4-6 P.M. (Mountain Time Zone) or send check or money order.

We also have big list of other Commodore products currently in stock at incredible prices. We also have quite a few Amiga and IBM items available. Send $1 for complete list and/or call for more information:

Computer Bargain Store
3366 South 2300 East
Salt Lake City, Utah 84109

Phone (801) 466-8084.

FGM Connection Ad

There are no limits to what you can create and print with the FUN GRAPHICS MACHINE.

Just to list a few:

CALENDARS

POSTERS

BUSINESS CARDS

BUSINESS FORMS

CUSTOM LABELS

CHARTS

CUSTOM CHECKS

LETTER HEADS

GREETING CARDS

BROCHURES

NEWS LETTERS

DISK ENVELOPES

KEYBOARD OVERLAYS

and much much more!

The FUN GRAPHICS MACHINE.....$24.95

FGM FONT DISK.....$ 5.00

FGM GRAPHICS DISK.....$ 5.00

```
FGM
CLIP
ART DISK
1 .............$ 8.00

FGM
CLIP
ART
DISK 2 .............$ 5.00

FGM
CALENDAR
TEMPLATE
DISK.......$ 5.00

FGM
OVERLAY
TEMPLATE
DISK........$ 5.00

Also available is a DEMO DISK so you may see what the FUN GRAPHICS
MACHINE is all about.....................$ 2.00

Please add $3.50 for shipping and handling.


        The FGM CONNECTION
        P.O. BOX 2206
        ROSEBURG, OR.,97470

        Phone (503)-673-2234
```

(Layout and line breaks exactly as they appear on disk. –MS)